

Algorithmic Trading

Arthur Li

October 8, 2024

Contents

1	Introduction	1
1.1	Systematic Investments Overview	1
1.1.1	Alpha models	2
1.1.2	Risk Models	3
1.1.3	Transaction Cost Models	4
1.1.4	Portfolio Construction Models	4
1.1.5	Execution Model	5
1.1.6	Research	6
1.1.7	Risk Assessment	7
1.2	Data Analysis	8
1.2.1	Data Taxonomy	8
1.2.2	Financial Data Structures	12
1.2.3	Data Labelling Techniques	16
1.2.4	Data Sample Weights	18
1.2.5	Fractionally Differentiated Features	20
2	Mathematical Primer	22
2.1	Time Series Analysis	22
2.1.1	Stationary Time Series	22
2.1.2	Univariate Time Series Models	23
2.2	Classical Machine Learning	25
2.2.1	Ensemble Methods	25
2.3	Deep Learning	26
2.3.1	Deep Feedforward Networks	26
2.3.2	Regularisation for Deep Learning	30
3	Market Microstructure	32
3.1	Market Fundamentals	32
3.1.1	Liquidity Access in Equity Markets	32
3.1.2	Trading Mechanisms	33
3.1.3	Market Microstructure Primer	34
4	Equities Trading	35
5	Fixed Income Trading	36
6	Derivatives Trading	37
6.1	Fundamentals of the Market	37
6.1.1	Forward, Futures, and Options	37
6.1.2	Clearing House	38
6.1.3	OTC Markets	38
6.2	Forwards and Futures	41
6.2.1	Pricing	41
6.2.2	Hedging with Futures	42
6.2.3	Interest Rate Futures	44
7	Currency Trading	45
8	Commodities Trading	46
9	Appendix	47
9.1	Visual Studio Code	47
9.1.1	Setting Up Python	48
9.1.2	Debugging	48

1 Introduction

Based on the book by Rishi K. Narang (2013), Raja Velu (2020), Marcos Lopez Prado (2018) etc.

1.1 Systematic Investments Overview

A schematic of a live 'production' trading strategy is shown below, but does not include everything else necessary to create the strategy (i.e., research tools).

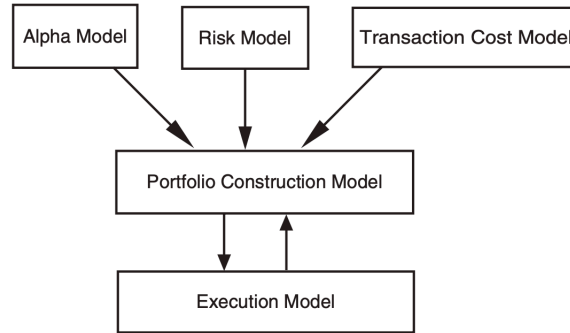


Figure 1: Live 'production' trading strategy

The trading system has three modules:

- i. Alpha model: predicts the future of the instruments considered for trading, i.e. directional alpha
- ii. Risk model: limits amount of exposure to factors that are unlikely to generate returns but could drive losses, i.e. directional exposure limit on an asset class
- iii. Transaction cost model: determine if the cost of the trades needed to migrate from current portfolio to new portfolio is desirable to the portfolio construction model.

These models feed into a portfolio construction model that balances the tradeoffs of profit and risk to determine the best portfolio to hold. The model finds the differences in trades that need to be executed.

The execution model then takes the required trades, and using inputs such as urgency in which the trades need to be executed and dynamics of liquidity in the markets, executes the trades in an efficient and low cost manner.

Method 1.1.1. *Chains of Production for Alpha Signals*

- i. Data Curation: for collecting, cleaning, indexing, storing, adjusting, and delivering all data to production chain. Requires experts in market microstructure and data protocols such as FIX.
- ii. Feature Analysis: transform raw data into informative signals. Requires experts in information theory, signal extraction and processing, visualisation, labelling, weighting, classifiers, feature importance techniques. Feature analysts collect and catalogue libraries of findings.
- iii. Strategists: informative features are transformed into actual investment algorithms. Strategists will parse libraries of features for ideas to develop an investment strategy. Require data scientists with deep knowledge of financial markets and economy. Features may be discovered by black box, but strategy is developed in a white box.
- iv. Back-testers: assess profitability of investment strategy under various scenarios. Requires data scientists with deep understanding of empirical and experimental techniques. Good back-tested incorporates in analysis meta-information on how strategy was created.
- v. Deployment Team: integrate strategy code into production line. Requires algorithm specialists and mathematical programmers. To ensure deployed solution is logically identical to prototype, and to optimise implementation sufficiently such that production latency is minimised.
- vi. Portfolio Oversight: once strategy is deployed, follows lifecycle.
 1. Embargo: initially, strategy is run on data observed after end date of backtest. If embargoed performance is consistent with backtest, strategy is promoted to next stage.
 2. Paper Trading: strategy run on live, real-time feed. Performance accounts for data parsing latencies, calculation latencies, execution delays, and other time lapses between observation and positioning.

3. Graduation: strategy manages real position, whether in isolation or as part of ensemble. Performance evaluated precisely, including attributed risk, returns, and costs.
4. Re-allocation: based on production performance, allocation is re-assessed frequently and automatically. Strategy allocation follows a concave function, Initial allocation is small. As time passes and strategy performs as expected, allocation is increased. Over time, performance decays and allocations become gradually smaller.
5. Decommission: if strategy perform below expectations for sufficiently extended period of time, strategy is discontinued.

1.1.1 Alpha models

Theory-driven models tests theories of why markets behave in a manner, and see if they can be used to predict the future. Strategies utilising price-related data are trend and mean reversion; strategies utilising fundamental data are value/yield, growth and quality. Usually more than one model is used in combination.

Definition 1.1.2. *Theory Driven Models*

- i. Trend Following: markets move in given direction long enough that the trend can be identified. As more data support the bull/bear thesis in an uncertain market, more market participants will adopt the same thesis and hence move the asset price to a new equilibrium.
Moving average crossover indicator strategy has less than one point of return for every point of downside risk taken, as market behaviour are unstable and episodic.
- ii. Mean Reversion: markets move in opposite direction to the prevailing trend. Short-term imbalances between buyers and sellers due to liquidity forces prices to move abruptly in one direction, which increases probability of trend reversion as liquidity issue is resolved.
Statistical arbitrage bets on convergence of prices of similar stocks whose prices have diverged.
Longer-term trends can occur despite smaller oscillations around these trends occurring in the shorter term, hence both strategies may be used in conjunction.
- iii. Value/Yield: value strategies uses ratios of fundamental factor against the price of the instrument, inverted to keep the ratio consistent. The higher the yield, the cheaper the instrument.
Buying undervalued security and selling overvalued security is a *carry trade*. The difference between yield received and yield paid is the *carry*.
Quant Long Short (QLS) ranks stocks by attractiveness based on various factors such as value, then buy the higher-ranked stocks while shorting the lower-ranked stocks.
- iv. Growth: make predictions based on asset's expected or historically observed level of economic growth. Forward-looking growth expectations are typically used as a metric.
Growth is trending, and strongest growers are becoming more dominant relative to competitors. Macro growth factors may be used on foreign exchange, while micro growth factors may be used on companies.
- v. Quality: All else being equal, it is better to long high quality and short low quality. Capital safety is important. Factors include earnings quality, equity-to-debt ratios etc.

Data-driven models are more difficult to understand, with more complicated mathematics. Relies on data mining, more technically challenging and far less widely practiced. Typically more used in high-frequency space, as they can discern how market behaves without caring about the economic theory or rational.

Method 1.1.3. *Strategy Parameters*

An implementation approach requires a forecast target, time horizon, bet structure, investment universe, model specification, and run frequency.

- i. Forecast Target: models may forecast direction, magnitude, duration of move, and may include probability into the forecast. Signal strength is of importance, defined by a larger expected return and/or higher likelihood of return. A higher level of signal strength results in a bigger bet taken on the position.
- ii. Time Horizon: models may have forecast horizons ranging from microseconds to years. There are more variability between short-term and long-term strategies, as short-term strategies are making very large number of trades compared to long-term strategies.
- iii. Bet Structure: models can be made to forecast an instrument relative in itself or to others. For relative forecasts, smaller clusters (pairs) or larger clusters (sectors) may be used. For pairs, few assets can be compared precisely and directly. Large cluster grouping may eliminate impact of general movement of the sector and hence focus on the relative movement of stocks within the sector, allowing for clearer distinction between group behaviour and idiosyncratic behaviour. Clusters may be created either via statistical methods or using heuristics (i.e., fundamentally defined industry groups).

Statistical methods may be fooled by data, leading to bad grouping. Heuristic grouping may be imprecise for conglomerates, and may be too rigid. Relative alpha strategies tend to exhibit smoother returns during normal times than intrinsic alpha strategies, but may face incorrect groupings during stressful periods. This may be mitigated by utilising several grouping techniques in concert.

- iv. Investment Universe: choices made on geography, asset class, instrument class, and exclusions. Liquidity is preferred so estimations of transaction costs are reliable. Large quantities of high quality data is required, which is found in highly liquid and developed markets. Instruments with consistent behaviour is preferred, hence biotech stocks are excluded due to sudden, violent price changes. Hence, the most common asset classes and instruments modelled are common stocks, futures (on bonds and equity indices) and forex.
- v. Model Specification: focuses on definition of the strategy mathematically, and may be the source of alpha. Specification details in terms of machine learning or data mining techniques are also defined, to assist in fitting models to the data and setting parameter values. Refitting frequency is also defined to refresh the model and make it adapt to current market conditions; may lead to greater risk of overfitting.
- vi. Run Frequency: defined from monthly to real time frequency. Increasing frequency of runs lead to greater number of transactions and hence higher transaction costs, and risk of moving portfolio based on noisy data. Less frequency of runs lead to smaller number of larger-sized trades, hence may move the market with block trades; may also miss opportunities to trade at more favourable prices.

Method 1.1.4. *Blending of Models*

Most common approaches are linear models, nonlinear models, and machine learning models. If models are not combined, then several portfolios are constructed based on output from each model, then combined using portfolio construction techniques. The best method depends on the model.

- i. Linear Models: require independence of factors, and each factor to be additive. To determine the weight of each alpha factor, multiple regression techniques may be used.
- ii. Nonlinear Models: used when factors are not independent, or the relationship changes over time. Conditional models base the weight of one factor on the reading of another factor. Rotational models assign weights of factors that fluctuate over time based on updated calculations of the various signal's weights, giving higher weights to factors with better performance recently.
- iii. Machine Learning Models: developing machine learning strategies takes as much effort to produce one true investment strategy as to produce a hundred. The complexities include data curation and processing, HPC infrastructure, software development, feature analysis, execution simulations, backtesting etc. Decades ago, macroscopic alpha based on simple tools like econometrics are common, but this is quickly diminishing. Microscopic alpha however, becomes more abundant, but requires heavy ML tools.

1.1.2 Risk Models

Risk model concerns the intentional selection and sizing of exposures to improve the quality and consistency of returns. By pursuing an alpha, we want to be invested in the movement of the exposure to profit in the long run.

Method 1.1.5. *Limiting Size of Risk*

The quantitative risk models that limit the size of risk varies by the manner in which size is limited, how risk is measured, and what is having its size limited.

Size limits can be limited by hard constraints and penalties. A hard limit may be arbitrary, hence penalty functions may be built to allow a position to increase beyond the limit level, only if the alpha model expects a significantly larger return. The levels of limits and penalties may be determined from either theory or data.

To measure risk, there are two methodologies. The first is longitudinal, and measures risk through the volatility of an instrument. The second is to measure the correlation or covariance between assets (dispersion).

Size limiting may be applied to single positions and groups of positions (sectors, asset classes). It may also be applied to various types of risks and the amount of portfolio leverage.

Method 1.1.6. *Limiting the Types of Risk*

To eliminate unintentional exposure as there is no expectation of being compensated sufficiently for accepting them. This can be achieved through theoretical or empirical risk models.

- i. Theory-Driven Risk Models: focuses on systematic risk factors, derived from economic theory. Systematic risks cannot be diversified away. Equity may have market risk, sector risk, market capitalisation risk etc. Fixed income may have interest rate risk.
- ii. Empirical Risk Models: uses historical data to determine the unnamed systematic risks that should be measured and mitigated. Uses principal component analysis (PCA) to discern unnamed systematic risks that may correspond to named risk factors. Used by statistical arbitrage traders who are betting on exactly the component of an asset's return not explained by systematic risks.

1.1.3 Transaction Cost Models

Trade is made only if it increases the odds or magnitude of return (from alpha model), or if it decreases the odds or magnitudes of loss (from risk model). However, this improvement should be higher than cost of trading. The transaction cost model is not designed to minimise cost of trading, only to inform portfolio construction engine the cost of making any given trade.

Remark 1.1.7. *Transaction Cost Components*

- i. Commissions and Fees: paid to brokerages (access to other market participants), exchanges (improved transaction security) and regulators (operational infrastructure) for the services provided. The bank's infrastructure is used by quants, where the brokerage commissions are rather small on a per-trade basis. Brokers also collect clearing and settlement fees. Clearing is the activity involving regulatory reporting and monitoring, tax handling, and handling failure, taken place in advance of settlement. Settlement is the delivery of securities in exchange for payment in full.
- ii. Slippage: the change in price between the time the quant system decides to transact and the time when the order is at the exchange for execution. Trend-following strategies suffer most from slippage as assets are already moving in desired direction; mean-reverting strategies suffer the least from slippage. The lower the latency to market, the smaller the slippage. The more volatile an asset, the bigger the slippage.
- iii. Market Impact: measures how much an order moves the market by its demand for liquidity. The impact of the trade on the market is unknown until the trade has already been completed. There may also be interaction between slippage and market impact (i.e., selling when a stock is trending upwards).

Definition 1.1.8. *Types of Transaction Cost Models*

- i. Flat Model: cost of trading is the same, regardless of size of order. Model is reasonable if size traded is nearly always about the same, and liquidity remains sufficiently constant.
- ii. Linear Model: cost of trading increases at a constant rate relative to size of order. Better estimate than flat transaction cost model.
- iii. Piece-Wise Linear Model: using piece-wise linear functions to model costs. Balance between simplicity and accuracy; better accuracy than flat or linear models.
- iv. Quadratic Model: most computationally intensive, but also most accurate.

1.1.4 Portfolio Construction Models

Comes in two major forms: rule-based, optimisers. Rule-based models are based on heuristics, can be exceedingly simple or rather complex, and derived from human experience (trial and error). Optimisers comprises of an objective function and uses algorithms to reach the end goal.

Definition 1.1.9. *Rule-Based Models*

- i. Equal Position Weighting: used if portfolio manager believes that if a position is good enough to own, no other information is needed in determining its size. Strength of signal is not used as input in weighting. Model assumes that there is sufficient statistical strength and power to predict not only direction but also magnitude relative to other forecasts in the portfolio. Portfolio takes few large bets on 'best' forecast, many smaller bets on less dramatic forecasts; may take excess risk in an idiosyncratic event on a seemingly attractive position, resulting in adverse selection bias.
- ii. Equal Risk Weighting: adjust position sizes inversely to volatilities or a measure of risk. More volatile positions given smaller allocations, less volatile positions given larger allocations. When unit of risk is equalised, it is almost always a backward-looking measurement such as volatility. If volatility changes with time, then model will be misled.
- iii. Alpha-Driven Weighting: position size based primarily on alpha model. Alpha signal determines size of position, but usually with size limits. Constraints used also includes limits on size of total bet on a group. May also have a function that relates the magnitude of forecast to size of position. If model used in futures trend following, might suffer sharp drawdowns. Reliance on accuracy of alpha.
- iv. Decision-Tree Weighting: decision path to arrive at the allocation for given instrument, depending on type of alpha model and type of instrument. Constraints may include percentage limits for allocation. Model size grows dramatically if more alpha models or more types of positions are included.

Remark 1.1.10. *Optimisers Models Parameters*

Harry Markowitz's mean variance optimisation (MVO) as the pioneer model. Models are based on principles of modern portfolio theory (MPT). Inputs include asset expected return (mean), asset variance, expected

correlation matrix. Other inputs include size of portfolio in currency terms, desired risk level (volatility or expected drawdown), and other constraints such as liquidity, universe limits.

Model uses an objective function and an algorithm to seek the goal, usually maximising return of portfolio relative to volatility of portfolio returns.

- i. Expected Return: alpha models as basis of expected return, which also includes expected direction.
- ii. Expected Volatility: stochastic volatility forecasting methods is commonly used, as volatility may have high and low periods, with occasional jumps. GARCH model is most used.
- iii. Expected Correlation: as instrument correlations are not stable over time, it is more appropriate to group assets together before computing correlation within the group.

Method 1.1.11. *Optimisation Techniques*

- i. Unconstrained Optimisation: most basic form with no constraints. Might provide a single-instrument portfolio, where all money will be invested in instrument with highest risk-adjusted return.
- ii. Constrained optimisation: constraints include position limits, limits on various groupings of instruments. Might result in constraints driving the portfolio construction more than the optimiser.
- iii. Black-Litterman Optimisation: blends investor expectations with a degree of confidence about those expectations, and these with historical precedent evident in the data. Adjusts historically observed correlation levels by utilising investor's forecast of return for the various instruments.
- iv. Grinold and Kahn's Approach: builds a portfolio of signals, instead of sizing positions. To build factor portfolios, each of which are usually rule-based portfolios based on a single type of alpha forecast. Each portfolio backtested, then series of returns are then treated as instruments of a portfolio by the optimiser. Number of factor portfolios is more manageable, usually not more than 20. What is optimised is then a handful of factor portfolios. The model allows for inclusion of risk model, transaction cost model, portfolio size, and risk targets as inputs.
- v. Resampled Efficiency: to improve the inputs to optimisation by addressing oversensitivity to estimation error. To resample data using Monte Carlo simulation to reduce estimation error in inputs to the optimiser.
- vi. Data-Mining Approaches: machine learning techniques such as supervised learning or genetic algorithms used, as MVO involves searching many possible portfolios to find the best.

1.1.5 Execution Model

Two basic ways to execute trade: through electronic, or through human intermediary. For electronic execution, achieved through direct market access (DMA), which allows traders to utilise the infrastructure and exchange connectivity of brokerage firms to trade directly on electronic markets.

Execution algorithms can be acquired through building, using broker's, or a third-party software vendors.

Brokerages offer portfolio bidding, where the 'blind' portfolio for transaction is described by characteristics such as valuation ratios of longs and shorts, sector breakdown, market capitalisation etc. Broker then quote a fee in basis points in terms of the gross market value of portfolio traded. Hence, certainty is provided by the broker to the trader. Once agreement reached, broker receives fee and assumes risk of trading out the portfolio at future market prices, which may be better or worse than prices guaranteed.

Remark 1.1.12. *Order Execution Algorithm Parameters*

- i. Aggressive vs Passive: algorithm make decision of passive vs aggressive order, depending on how immediately the trader wants to do the trade. Market orders are considered aggressive. Limit order at current best order is fairly aggressive, while limit order below current bid is passive.
Many exchanges pay providers of liquidity for placing passive orders, charging traders for using liquidity provided. Orders that cross the spread are using liquidity by using a passive order placed by another trader, reducing liquidity available. Paying for liquidity sweetens deal for passive order, only if order is actually executed; passive trader gets better transaction price and a commission rebate from the exchange. Momentum strategies uses more aggressive orders; mean reversion uses more passive orders. A stronger, more certain signal will be executed with greater aggressiveness than a weaker or less certain signal. A middle ground will be to put limit orders between best current bid and offer.
- ii. Large vs Small Order: a large order may be broken into many smaller orders over a window of time, but risk price moving in adverse direction. Size of chunk depends on transaction cost model estimate, and analysis of correct level of aggressiveness.

- iii. Hidden vs Visible Order: a queue as a visible order gives away a bit of information. Hidden order will provide no information to the market, staving off imbalances, but reduces priority of trade in the queue. Algorithmic trading utilising hidden order is 'iceberging', which is taking a single larger order and chopping it into many smaller chunks, most posted to order book as hidden orders.
- iv. Order Routing: if there are several pools of liquidity for the same instrument, smart order routing will be used, which determines which pool of liquidity is most suitable for sending a given order. Depth of liquidity on various ECNs and connectivity speeds are also considered in smart order routing.
- v. Cancelling and Replacing Orders: traders may place larger number of orders with no intention of execution, then rapidly cancelling them and replacing them with other orders. This allows gaining of information on how market responds to the changing depth of the book, providing information on how to profit from the pattern of reaction. If trader wants to buy a large number of shares, he may enter a large number of small orders to sell the shares further away from market and cancel, improving market perception.

Definition 1.1.13. *High Frequency Trading*

Alpha driving strategies on extremely near-term bets (seconds or less) are *microstructure alphas*, focusing on liquidity patterns in order book. Larger quants may also use this to guide execution models, improving costs of entering trades. Small differences over a single trade add up significantly in the long run. To trade microstructure alpha as independent high frequency strategies, large investments in infrastructure and research must be done. Machine learning techniques may also be used to discern patterns in execution of other player orders. The more inferior the execution models, the easier it is to discern the pattern, allowing the ML strategy to profit from these patterns in the future. Patterns in the shorter timescale are somewhat stable.

Definition 1.1.14. *HFT Shark Strategy*

Designed to detect large orders that are icebergged, by sending series of very small trades; if each of these small orders get filled quickly, this may be a sign of a large and icebergged order. The shark simply front-run this large, hidden order by placing visible trades in front of the icebergged order. The iceberg strategy must then push prices up to execute trades. When the icebergged order is complete, prices will be pushed up favourably for the shark, which can then exit the position with a quick and relatively riskless profit.

Remark 1.1.15. *HFT Trading Infrastructure*

Using a broker that act as trading agent allows the infrastructure requirements to be handled by the broker, instead of dealing with the regulatory and other constraints.

High frequency strategies may use colocation or sponsored access. Colocation setup is where trader attempts to place trading servers as physically close to the exchange as possible.

Financial Information eXchange (FIX) protocol is the choice of real-time electronic communication among users. The software that implements the FIX protocol is free and open source (FIX engine). High frequency traders will likely build their own FIX engines to ensure optimal speeds.

1.1.6 Research

Definition 1.1.16. *Scientific Method*

1. Researcher observe a phenomenon in the market and construct a theory.
2. Researcher seeks out information to test the theory.
3. Researcher tests the theory, and with enough confidence, risk some capital on the validity of the theory.

Remark 1.1.17. *Sources of Alpha Idea Generation*

1. Observing the market, using the scientific method to test the theory
2. Academic literature, requiring significant time to read academic journals, working papers, and conference presentations for ideas. Literature from other fields such as astronomy, physics, or psychology, may provide ideas relevant to quant finance problems.
3. Migration of a researcher or portfolio manager from one quant shop to another.
4. Lessons from activities of discretionary traders

Remark 1.1.18. *Model Quality Assessment*

- i. Cumulative profit graph: if profit profile is not smooth, with long periods of inactivity, sharp losses and gains, then the model may have issues
- ii. Average annual rate of return: indicates how well the strategy made on historical data

- iii. Variability of returns: the less variable the level of returns, the better the strategy. May look at lumpiness of returns, which is the portion of strategy's total returns that comes from periods that are significantly above average (measures consistency of returns).
- iv. Worse Peak-to-Valley Drawdowns: measures maximum decline from any cumulative peak in profit curve. The lower the drawdown the better the strategy. Also, to measure recovery period after drawdowns; the shorter the recovery period the better the strategy.
- v. Predictive Power: R-squared statistic may be used, which shows how much of the variability of the predicted asset have been accounted for. A exceedingly high R^2 in would be 0.05 out of sample. Instrument returns may be bucketed by deciles; a model with reliable predictive power is one that appropriately buckets the instruments correctly.
- vi. Percentage Winning Trades, Winning Time Periods: whether the strategy tends to make profits from a small portion of trades that do very well, or from a large number of trades.
- vii. Ratios of Returns vs Risk: Statistics such as risk-adjusted return, Sharpe ratio, information ratio, Sterling ratio, Calmer ratio, Omega ratio.
- viii. Relationship with Other Strategies: value-add of new strategy compared with results of existing strategy with and without the new idea.
- ix. Time decay: understand strategy returns if trades are initiated on lagged basis after receiving a trading signal. Determine strategy sensitivity to timeliness with information received, and crowdedness of strategy.
- x. Sensitivity to specific parameters: high quality strategy has small changes in outcomes from slight changes in parameters. Or else this may be a sign that model may be overfitted.
- xi. Overfitting: plot a graph of parameter value vs function outcome; a good model has a flatter curve with no jumps. Models that are parsimonious (less parameters) uses less assumptions, hence less overfitting.

Remark 1.1.19. *Other Considerations in Model Testing*

Overestimation of trading costs may cause portfolio to hold positions for longer than optimal, and underestimation may result in high portfolio turnover and bleed from trading costs. Assumptions on availability of short positions must also be made; hard-to-borrow lists must be taken into consideration.

1.1.7 Risk Assessment

Definition 1.1.20. *Model Risks*

Quant models has model risk, the risk that the model does not accurately describe, match, or predict the real-world phenomenon. Each component of the quant model may all have model risk.

- i. Inapplicability of Modelling: occurs when quant model is mistakenly applied to a problem. May also occur with misapplication of a technique to a given problem.
- ii. Model Misspecification: occurs when the model doesn't fit the real world. Model may work fine most of the time, but fail when an extreme event occurs.
- iii. Implementation Errors: errors in programming or architecting systems. Architectural error may also occur when models are loaded in a wrong sequence.

Definition 1.1.21. *Regime Change Risk*

Quant models are based on relationships prevalent in historical data. If there is a regime change, the historical relationships and behaviour may be altered, hence the model may lose effectiveness.

Definition 1.1.22. *Exogenous Shock Risk*

Risks driven by information that is not internal to the market, i.e., terrorist attacks, start of wars, bank bailouts, change in regulation such as in shorting rules. May require discretionary overrides.

Definition 1.1.23. *Contagion Risk*

Happens when other investors hold the same strategies. First part of risk factor relates to how crowded the quant strategy is. Second part relates to what else is held by other investors that could force them to exit the quant strategy in a panic (ATM effect).

Quant liquidation criss may be driven by size and popularity of quantitative strategies, subpar returns from operators leading up to the crisis, the practice of funds cross-collateralising many strategies against each other, and risk targeting (risk managers target a specific level of volatility for their funds or strategies).

Method 1.1.24. *Risk Monitoring Methods*

- i. Exposure Monitoring Tools: with current positions held, the positions are grouped for the various exposures (i.e., valuation, momentum level, volatility) to monitor gross and net exposure to various sectors and industries, buckets of market capitalisation, various style factors.

- ii. Profit and Loss Monitors: with current portfolio, compare that with previous day closing price. Intraday performance charts are used. May also look at source of profit, hit rate (percentage of time strategy makes money on a given position).
- iii. Execution Monitors: shows progress of executions, i.e., which orders are currently being worked on, which ones are completed, with transaction size and prices. Fill rates for limit orders are used for more passive execution strategies. Slippage and market impact are also monitored.
- iv. System Performance Monitors: checks for software and infrastructure errors. Checks performance of CPUs, speeds of various stages of automated processes, latency in communication of messages.

1.2 Data Analysis

1.2.1 Data Taxonomy

A quick overview of data used in systematic trading.

Four essential types of financial data

Fundamental Data	Market Data	Analytics	Alternative Data
<ul style="list-style-type: none"> • Assets • Liabilities • Sales • Costs/Earnings • Macro Variables • ... 	<ul style="list-style-type: none"> • Price/Yield/IV • Volume • Dividend/Coupons • Open Interest • Quotes/Cancellations • Aggressor Side • ... 	<ul style="list-style-type: none"> • Analyst Recommendation • Credit Ratings • Earnings Expectations • News Sentiment • ... 	<ul style="list-style-type: none"> • Satellite/CCTV • Google Searches • Twitter/Chats • Metadata • ...

Remark 1.2.1. *Fundamental Data Characteristics*

- i. Data published is indexed by last date included in report, which precedes date of release.
- ii. Data is often backfilled or re-instated, and data vendor may overwrite initial values with corrections.
- iii. Data is extremely regularised and low frequency.

Remark 1.2.2. *Market Data Characteristics*

- i. Raw feed contains unstructured information, such as FIX messages (allow full construction of trading book), or full collection of BWIC (bids wanted in competition) responses.
- ii. FIX data is not trivial to process, $\sim 10\text{TB}$ generated on daily basis

Remark 1.2.3. *Analytics Data Characteristics*

- i. Derivative data as processed based on original source. Signal already extracted from the original source.
- ii. Costly, methodology used in production may be biased or opaque.

Remark 1.2.4. *Alternative Data Characteristics*

- i. Produced by individuals, business processes, and sensors.
- ii. Primary information that has not made it to other sources.
- iii. Cost and privacy concerns. May be useful if it annoys data infrastructure team.

Definition 1.2.5. *Reference Data*

- i. Trading Universe: evolving daily to incorporate new listings, de-listings etc. Knowing when a particular stock no longer trades is important to avoid survivor bias.
- ii. Symbology Mapping: ISIN, SEDOL, RIC, Bloomberg Tickers etc. Data is not static, symbols may change, complicating historical data merges. Mapping needs to persist as point-in-time data and allow for historical 'as-of-date' usage, require implementation of bi-temporal data structure.
- iii. Ticker Changes: for reasons described in symbology mapping. To maintain historical table of ticker changes to seamlessly go up and down time series data.
- iv. Corporate Actions Calendars: contain stock and cash dividends (announcement, execution date), stock splits, reverse splits, rights offer, mergers and acquisitions, spin off, free float or shares outstanding adjustments, quotation suspensions etc.
For dividends, announcements may coincide with more volatility, jumps in price time series. Allow building of strategies that look to benefit from the added volatility.

For stock splits, reverse splits, rights offers, all historical data need to be adjusted backward to reflect the split (both volume and price).

For M&A, spin-offs, to account for changes in valuation, hence used in Merger Arbitrage strategies.

Suspensions result in gaps in data, may impact backtesting.

- v. Static Data: country, sector, primary exchange, currency, and quote factor. May be used to group instruments based on fundamental similarities (hence for pairs trading). Maintaining a table of quotation currency per instrument necessary to aggregate positions at portfolio level.
- vi. Exchange Specific Data: individual exchanges have variety of differences to be accounted for when designing trading strategies. First group of information concerns the hours and dates of operations:
 - 1. Holiday Calendar: Strategies trading simultaneously in several markets and leveraging correlation may not perform as well if one market is closed and another is open.
 - 2. Exchange Sessions Hours: Different sessions (Pre-Market, Continuous Core, After-Hour etc.); auction times and respective cutoff times for order submission; lunch break restricting intraday trading and auctions before/after lunch; settlement times for futures market. Daylight Saving Time (DST) adjustments; length of trading hours during course of the year; different trading hours by venue.
 - 3. Disrupted Days: Exchange outages or trading disruptions, market data issues. To be recorded so they can be filtered out when building or testing strategies.

Second group of information governing the mechanics of trading:

- 1. Tick Size: Minimum eligible price increment; may vary by instrument and as a function of price.
 - 2. Trade and Quote Lots: Minimum size increment for quotes or trades.
 - 3. Limit-Up and Limit-Down Constraints: Maximum daily fluctuations of securities, and whether trading is paused or can only be traded at better prices than the threshold.
 - 4. Short Sell Restrictions: Restrict short sells not to trade at price worse than last price, or not to create a new quote that will be lower than the lowest prevailing quote. Impact ability to source liquidity.
- vii. Market Data Condition Codes: vary per exchange and asset class, and each market event may be attributed to several codes at once. To build mapping table of condition codes and what they mean (i.e., auction trade, lit or dark trade, cancelled or corrected trade, regular trade, off-exchange trade reporting, block-size trade, trade originating from multi-leg order such as option spread trade etc.). To access liquidity for trading algorithm, trades published for reporting purposes must be excluded and not be used to update some of the aggregated daily data used in construction of trading strategies.
- viii. Special Day Calendars: days with distinct liquidity characteristics to be accounted for in both execution strategies and in alpha generation process. These (non-exhaustive) irregular events may be:
 - 1. Half trading days preceding Christmas and following thanksgiving in US
 - 2. Ramadan even in Turkey
 - 3. Taiwan market opening on weekend to make up for lost trading days during holiday periods
 - 4. Korean market changing trading hours on day of nationwide university entrance exam
 - 5. Brazilian market opening late on day following the Carnival
 - 6. Last trading days of months and quarters (investors rebalance portfolios)
 - 7. Index rebalancing dates, where intraday volume distribution is significantly skewed toward EODs
 - 8. Options and futures expiry dates (quarterly/monthly expiry, Triple Witching in US, Special Quotations in Japan) where excess trading volume and different intraday patterns result from hedging activity and portfolio adjustments.

Model normal days first. Special days are modelled either independently, or using normal days as baseline.

- ix. Futures-Specific Reference Data: to know which contract was live at any point of time by using expiry calendar, and the most liquid contract. Equity index futures are most liquid for first contract available (front month), energy futures such as oil are more liquid for second contract. Hence to know which contract carry the most significant price formation characteristics, and what is true liquidity available. Note there is no real standardised expiry frequency that applies across markets. When computing rolling-window metrics, to account for potential roll dates (due to investors rolling forward positions) that may have happened during the time span. May blend volume time series prior to roll date and after roll date. Futures market also have different market phases during the day with significantly different liquidity characteristics. Various market data metrics (volume profile, average spread, average bid-ask sizes etc) should be computed separately for each market phase by maintaining a table of start and end times of each session for each contract.

- x. Options-Specific Reference Data (Options Chain): expiry date and strike price combination (option chain). Map of equity tickers to option tickers with strike and expiry dates allow for design for more complex investment and hedging strategies (i.e., distance to strike, change in open interest of puts and calls).
- xi. Market-Moving News Releases: macroeconomic announcements. To maintain calendar of dates and times of their occurrences to assess their impact on strategies. Central bank announcements or meeting minutes releases about major economies (FED/FOMC, ECB, BOE, BOJ, SNB), Non-Farm Payrolls, Purchasing Managers' Index, Manufacturing Index, Crude Oil Inventories etc.
Stock-specific releases such as earnings calendars, specialised sector events (for healthcare, biotech etc).
- xii. Related Tickers: tickers that are related to each other as they fundamentally represent the same underlying asset. Allows efficient opportunity exploitation. Primary tickers to composite tickers mapping (for markets with fragmented liquidity), dual listed/fungible securities in US and Canada, ADR or GDR, local and foreign boards in Thailand etc.
- xiii. Composite Assets: ETFs, Indexes, Mutual Funds etc. May be used to achieve desired exposures, or as cheap hedging instruments, and provide arbitrage opportunities when they deviate from NAV. To maintain information such as time series of their constituents and value of any cash component, divisor used to translate NAV into quoted price, constituent weights.
- xiv. Latency tables: for higher frequency trading strategies. Contains distribution of latency between different data centres for more efficient order routing, and reordering data that are recorded in different locations.

Definition 1.2.6. *Market Data Feed*

- i. Level I Data (Trade and BBO Quotes): trades and top of book quotes. Enough to reconstruct Best Bid and Offer (BBO). Also contains information in form of trade status (cancelled, reported late etc), trade and quote qualifiers (odd lot, normal trade, auction trade, Intermarket Sweep, average price reporting, on which exchange etc). May be used to analyse sequence of events and decide if a given print should be used to update the last price and total volume traded at a point in time.
- ii. Level II Data (Market Depth): addition of quote depth data, displays all lit limit order book updates (price changes, addition or removal of shares quoted) at any level in the book, for all of the lit venues in fragmented markets.
- iii. Level III Data (Full Order View): message data. Each order arriving is attributed a unique ID for tracking over time, and is precisely identified when it is executed, cancelled, or amended. Possible to build a full (with national depth) book at any moment intraday. Example from US market:
 1. Timestamp: number of milliseconds after midnight
 2. Ticker: equity symbol (up to 8 char)
 3. Order: Unique order ID
 4. T: message type. 'B' is add buy order; 'S' is add sell order; 'E' is execute outstanding order in part; 'C' is cancel outstanding order in part; 'F' is execute outstanding order in full; 'D' is delete outstanding order in full; 'X' is bulk volume for cross event; 'T' is execute non-displayed order
 5. Shares: order quantity for 'B', 'S', 'E', 'X', 'C', 'T' messages. Zero for 'F', 'D' messages
 6. Price: order price for 'B', 'S', 'X', 'T' messages. Zero for cancellation and executions. Last 4 digits are decimal, padded to right with zeroes. Divide by 1000 to convert to currency value.
 7. MPID: Market Participant ID associated with transaction (4 char)
 8. MCID: Market Centre Code for originating exchange (1 char)

A few special types of orders worth mentioning are:

1. Order subject to price sliding: execution price may be one cent worse than display price at NASDAQ; ranked at locking price as hidden order, displayed at the price one minimum price variation inferior to locking price. New order ID will be used if order is replaced as a display order.
2. Pegged order: based on NBBO, not routable, new timestamp given upon repricing; display rule vary over exchanges
3. Mid-point peg order: non-displayed, may result in half-penny execution
4. Reserve order: displayed size is ranked as displayed limit order; reserve size is behind non-displayed orders and pegged orders in priority.
Minimum display quantity is 100, amount replenished from reserve size when it falls below 100 shares; New timestamp created, displayed size re-ranked upon replenishment.
5. Discretionary order: displayed at one price while passively trading a more aggressive discretionary price. Order becomes active when shares are available within discretionary price range. Order ranked last in priority. Execution price may be worse than display price.

6. Intermarket sweep order: can be executed without need for checking prevailing NBBO.

Using these data, we may model: the pattern of inter-arrival times of various events; arrival and cancellation rates as a function of distance from nearest touch price; arrival and cancellation rates as a function of other information, such as in the queue on either side of the book, order book imbalance etc.

Once modelled, we may analyse: the impact of market order on limit order book; chances for limit order to move up the queue from given entry position; probability of earning the spread; expected direction of price movement over a short horizon.

Definition 1.2.7. *Binned Data*

- i. Open, High, Low, Close (OHLC) and Previous Close Price: indication on trading activity and intraday volatility. Distance traveled between lowest and highest points is indication of market sentiment. Previous close has to be adjusted for corporate actions and dividends.
- ii. Last Trade before Close (Price/Size/Time): how much the close price may have jumped in final moments of trading; how stable it is as a reference value for next day.
- iii. Volume: trading activity indicator, especially when level jumps from long term average. Collect volume breakdown between lit and dark venues for execution strategies.
- iv. Auctions Volume and Price: price discovery event when significant volume prints occur.
- v. VWAP: indication of trading activity on the day. Easier to algorithmically execute large orders with VWAP than a single print.
- vi. Short Interest/Days-to-Cover/Utilisation: good proxy for investor position. Short pressure an indication of upcoming short term moves: large short interest is bearish view from institutional investors. Utilisation level of available securities to borrow gives indication of how much room is left for future shorting. Days-to-Cover to assess magnitude of potential short squeeze (if sellers unwind position, fraction of available daily liquidity needed); larger value indicates larger potential of sudden upswing on heavily shorted securities.
- vii. Futures Data: insight into activity or large investors through open interest data. Offer arbitrage opportunities if their basis exhibits mis-pricing compared to one's dividend estimates.
- viii. Index-Level Data: source of relative measures for instrument specific features (index OHLC, volatility). Normalised features identify individual instruments deviating from their benchmarks.
- ix. Options Data: information on position of traders through open interest and Greeks.
- x. Asset Class Specific: yield/benchmark rates (repo, 2y, 10y, 30y), CDS spreads, US Dollar Index

Definition 1.2.8. *Granular Intraday Microstructure Activity*

- i. Number and Frequency of Trades: proxy for activity level, and how continuous it is. Low number of trades mean harder execution, and may be more volatile
- ii. Number and Frequency of Quote Updates: similar proxy for activity level
- iii. Top of Book Size; proxy for liquidity of instruments (larger top of book size makes it possible to trade larger order quasi immediately)
- iv. Depth of Book (price and size): similar proxy for liquidity
- v. Spread Size (average, median, time weighted average): proxy for cost of trading. Parametrised distribution used to identify opportunities if they are cheap or expensive
- vi. Trade size (average, median): to identify intraday liquidity opportunities when examining volume available in the order book.
- vii. Ticking time (average, median): representation of how often one should expect changes in the order book first level. For execution algorithms for which the frequency of updates (adding/cancelling child orders, re-evaluating decisions etc.) should commensurate with characteristics of the traded instrument.

Daily distribution can be used as start of day estimates and updated intraday with online Bayesian updates. Last group of daily data is derived from previous two groups but stored pre-computed to save time during research phase, or to be used as normalising values:

- i. X-day Average Daily Volume (ADV) / Average Auction Volume
- ii. X-day Volatility (close-to-close, open-to-close etc)
- iii. Beta with respect to index or sector (plain beta, or asymmetric up-days/down-days beta)
- iv. Correlation matrix

When binning data, this may be grouped into bins ranging from a few seconds to 30 minutes. Minute bar data are used for volume and spread profiles to prevent introducing excess noise due to market friction.

Definition 1.2.9. *Fundamental Data and Other Data*

- i. Key Ratios: Earnings Per Share (EPS), Price-to-Earning (P/E), Price-to-Book (P/B), etc.
- ii. Analyst Recommendations: aggregated values given consensus valuation
- iii. Earnings data: estimations by research analysts provide quarterly earning estimates which can be used as indication of performance of stock before actual value is published
- iv. Holders: sudden changes in ownership indicate changes in sentiment by sophisticated investor
- v. Insiders Purchase/Sale: indicator of future stock price moves from group of people who have access to best possible information about the company
- vi. Credit Ratings: credit downgrades resulting in higher funding costs have negative impact on equity prices

1.2.2 Financial Data Structures

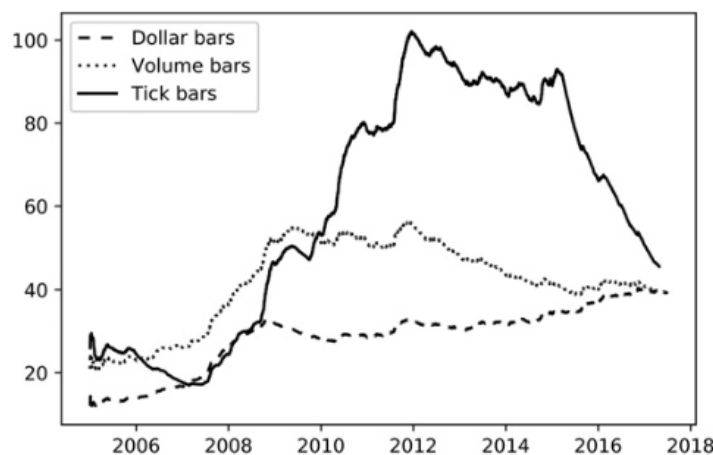


Figure 2: Average daily frequency of tick, volume, and dollar bars

Remark 1.2.10. *Standard BARS*

Method to transform a series of observations arriving at irregular frequency into a homogeneous series derived from regular sampling.

- i. Time Bars: obtained by sampling information at fixed intervals. Information collected includes timestamp, volume-weighted average price (VWAP), open price, close price, high price, low price, volume etc. To be avoided as markets do not process information at constant time interval. Time bars oversample information in low-activity periods and under-sample information in high-activity periods. Time bars exhibit poor statistical properties, i.e., serial correlation, heteroscedasticity, non-normality of returns.
- ii. Tick Bars: sample variables extracted each time a pre-defined number of transactions take place. Allows synchronisation of sampling with a proxy of information arrival. Sampling as a function of trading activity creates returns closer to IID Normal (Thierry and Helyette (2000)). When constructing tick bars, to be aware of outliers, as many exchanges carry out auction at open and at close; order book accumulates bids and offers without matching. Order fragmentation introduces some arbitrariness in number of ticks. Matching engine protocols may split one fill into multiple artificial partial fills as a matter of operational convenience.
- iii. Volume Bars: samples every time a pre-defined amount of security's units that have been exchanged. Achieves better statistical properties than sampling tick bars. Convenient artefact for studying market microstructure theories.
- iv. Dollar Bars: samples an observation every time a pre-defined market value is exchanged. Used when the analysis involves significant price fluctuations. Robust against corporate actions such as splits, reverse splits, issuance of new shares, buying back existing shares. Bar size could be dynamically adjusted as a function of free-floating market cap of a company or outstanding amount of issued debt.

Remark 1.2.11. *Information-Driven Bars*

Method to sample more frequently when new (micro-structural) information arrives to the market.

- i. Tick Imbalance Bars: sample bars whenever tick imbalance exceeds expectations. To determine tick index T such that accumulation of signed ticks exceeds a given threshold.
 Let $\{(p_t, v_t)\}_{t=1, \dots, T}$ be sequence of ticks where p_t and v_t is the price and volume associated with tick t .
 Let tick rule define a sequence $\{b_t\}_{t=1, \dots, T}$ where

$$b_t = \begin{cases} b_{t-1} & \text{if } \Delta p_t = 0 \\ \frac{|\Delta p_t|}{\Delta p_t} & \text{if } \Delta p_t \neq 0 \end{cases}$$

The tick imbalance at time T is defined as

$$\theta_T = \sum_{t=1}^T b_t$$

Compute expected value of θ_T at beginning of the bar,

$$E_0[\theta_T] = E_0[T](P[b_t = 1] - P[b_t = -1]) = E_0[T](2P[b_t = 1] - 1)$$

where $E_0[T]$ is expected size of tick bar, $P[b_t = 1]$ and $P[b_t = -1]$ is unconditional probability that a tick is classified as a buy and sell. In practice, $E_0[T]$ and $(2P[b_t = 1] - 1)$ may be estimated as an exponentially weighted moving average of T and b_t values from prior bars.

Define the tick imbalance bar (TIB) as a T^* contiguous subset of ticks such that

$$T^* = \arg \min_T \{|\theta_T| \geq E_0[T] | 2P[b_t = 1] - 1\}$$

where the size of expected imbalance is implied by $|2P[b_t = 1] - 1|$.

When θ_T is more imbalanced than expected, a low T will satisfy the conditions.

TIBs are produced more frequently under presence of informed trading (asymmetric information that triggers one-side trading). TIBs are buckets of trades containing equal amounts of information.

- ii. Volume/Dollar Imbalance Bars: sample bars when volume or dollar imbalances diverge from expectations. First, define imbalance at time T as

$$\theta_T = \sum_{t=1}^T b_t v_t$$

where v_t may represent ether number of securities traded (VIB) or dollar amount traded (DIB).

The expected value of θ_T at the beginning of the bar is then computed as

$$\begin{aligned} E_0[\theta_T] &= E_0 \left[\sum_{t|b_t=1}^T v_t \right] - E_0 \left[\sum_{t|b_t=-1}^T v_t \right] \\ &= E_0[T](P[b_t = 1]E_0[v_t|b_t = 1] - P[b_t = -1]E_0[v_t|b_t = -1]) \\ &= E_0[T](v^+ - v^-) \end{aligned}$$

where the initial expectation of v_t is decomposed into component contributed by buys and sells. Then

$$E_0[\theta_T] = E_0[T](2v^+ - E_0[v_t])$$

In practice, $E_0[T]$ and $(2v^+ - E_0[v_t])$ may be estimated as exponentially weighted moving average of T and $b_t v_t$ values from prior bars. Next, define VIB or DIB as a T^* -contiguous subset of ticks such that

$$T^* = \arg \min_T \{|\theta_T| \geq E_0[T] | 2v^+ - E_0[v_t]\}$$

where the size of expected imbalance is implied by $|2v^+ - E_0[v_t]|$.

When θ_T is more imbalanced then expected, a low T will satisfy the conditions.

VIB and DIB addresses concerns on tick fragmentation and outliers, and also addresses the issues of corporate actions, as the bar size is adjusted dynamically.

- iii. Tick Runs Bars: sample bars when the sequence of buys in overall volume diverges from expectations. For the case when large traders sweep order book, use iceberg orders, or slice parent orders into multiple

children, all leaving a trace of runs in the $\{b_t\}_{t=1,\dots,T}$ sequence. Define length of current run as

$$\theta_T = \max \left\{ \sum_{t|b_t=1}^T b_t - \sum_{t|b_t=-1}^T b_t \right\}$$

The expected value of θ_T at beginning of bar is computed as

$$E_0[\theta_T] = E_0[T] \max\{P[b_t = 1], 1 - P[b_t = 1]\}$$

In practice, $E_0[T]$ and $P[b_t = 1]$ may be estimated as exponentially weighted moving average of T and proportion of buy ticks from prior bars. Next, define TRB as T^* -contiguous subset of ticks such that

$$T^* = \arg \min_T \{\theta_T \geq E_0[T] \max\{P[b_t = 1], 1 - P[b_t = 1]\}\}$$

where the expected count of ticks from runs is implied by $\max\{P[b_t = 1], 1 - P[b_t = 1]\}$.

When θ_T exhibits more runs than expected, a low T will satisfy these conditions.

Instead of measuring length of longest sequence, count number of ticks of each side without offsetting.

- iv. Volume/Dollar Runs Bars: sample bars when volume or dollars traded by one side exceed expectation for a bar. First, define volume or dollars associated with a run as

$$\theta_T = \max \left\{ \sum_{t|b_t=1}^T b_t v_t - \sum_{t|b_t=-1}^T b_t v_t \right\}$$

where v_t may either represent volume (VRB) or dollar amount exchanged (DRB). The expected value of θ_T at beginning of the bar is then

$$E_0[\theta_T] = E_0[T] \max\{P[b_t = 1]E_0[v_t|b_t = 1], (1 - P[b_t = 1])E_0[v_t|b_t = -1]\}$$

In practice, $E_0[T]$, $P[b_t = 1]$, $E_0[v_t|b_t = 1]$, $E_0[v_t|b_t = -1]$ may be estimated as exponentially weighted moving average of T , proportion of buy ticks, buy volumes, and sell volumes from prior bars. Next, define a volume runs bar (VR) as T^* -contiguous subset of ticks such that

$$T^* = \arg \min_T \{\theta_T \geq E_0[T] \max\{P[b_t = 1]E_0[v_t|b_t = 1], (1 - P[b_t = 1])E_0[v_t|b_t = -1]\}\}$$

expected volume from runs is implied by $\max\{P[b_t = 1]E_0[v_t|b_t = 1], (1 - P[b_t = 1])E_0[v_t|b_t = -1]\}$.

When θ_T exhibits more runs than expected, volume from runs is greater than expected, a low T will satisfy these conditions.

Definition 1.2.12. *Multi-Product Series: ETF Trick*

To model a basket of securities as if it was a single cash product. To transform any complex multi-product dataset into a single dataset that resembles a total return ETF.

Method 1.2.13. *ETF Trick*

Produce a time series that reflects the value of \$1 invested. Changes in the series will reflect changes in PnL, series will be strictly positive, and implementation shortfall will be taken into account. The bars contain:

- i. Raw open price of instrument $i = 1, \dots, I$ at bar $t = 1, \dots, T$: $o_{i,t}$
- ii. Raw close price of instrument $i = 1, \dots, I$ at bar $t = 1, \dots, T$: $p_{i,t}$
- iii. USD value of one point of instrument $i = 1, \dots, I$ at bar $t = 1, \dots, T$: $\varphi_{i,t}$. This includes forex rate.
- iv. Volume of instrument $i = 1, \dots, I$ at bar $t = 1, \dots, T$: $v_{i,t}$
- v. Carry, dividend, or coupon paid by instrument i at bar t : $d_{i,t}$. Variable can also be used to charge margin costs or costs of funding.

All instruments $i = 1, \dots, I$ were tradable at bar $t = 1, \dots, T$. Even if some instruments were not tradable over entirety of time interval $[t - 1, t]$, at least they were tradable at times $t - 1$ and t .

For basket of securities with allocation vector ω_t rebalanced (or rolled) on bars $B \subseteq \{1, \dots, T\}$, the \$1 investment

value $\{K_t\}$ is derived as

$$h_{i,t} = \begin{cases} \frac{\omega_{i,t} K_t}{o_{i,i+1} \varphi_{i,t} \sum_{i=1}^I |\omega_{i,t}|} & \text{if } t \in B \\ h_{i,t-1} & \text{otherwise} \end{cases}$$

$$\delta_{i,t} = \begin{cases} p_{i,t} - o_{i,t} & \text{if } (t-1) \in B \\ \Delta p_{i,t} & \text{otherwise} \end{cases}$$

$$K_t = K_{t-1} + \sum_{i=1}^I h_{i,t-1} \varphi_{i,t} (\delta_{i,t} + d_{i,t})$$

where $K_0 = 1$ is the initial AUM. Variable $h_{i,t}$ is the holdings of instrument i at time t , $\delta_{i,t}$ is change of market value between $t-1$ and t for instrument i . Note profits or losses are being reinvested whenever $t \in B$, hence preventing negative prices. Dividends $d_{i,t}$ are already embedded in K_t .

The purpose of $\omega_{i,t} \left(\sum_{i=1}^I |\omega_{i,t}| \right)^{-1}$ is to de-lever the allocations.

Let τ_i be transaction cost associated with trading \$1 of the instrument. Three additional variables that the strategy needs to know for every observed bar t are:

- i. Rebalance Costs: variable cost $\{c_t\}$ associated with allocation rebalance is

$$c_t = \sum_{i=1}^I (|h_{i,t-1}| p_{i,t} + |h_{i,t}| o_{i,t+1}) \varphi_{i,t} \tau_i \quad \forall t \in B$$

Note c_t is not embedded in K_t , as shorting will generate fictitious proceeds when allocation is rebalanced. In code, $\{c_t\}$ is treated as a (negative) dividend.

- ii. Bid-Ask Spread: the cost $\{\tilde{c}_t\}$ of buying or selling one unit of this ETF,

$$\tilde{c}_t = \sum_{i=1}^I |h_{i,t-1}| p_{i,t} \varphi_{i,t} \tau_i$$

When a unit is bought or sold, strategy must charge this cost \tilde{c}_t .

- iii. Volume: volume traded $\{v_t\}$ is determined by least active member in the basket. Let $v_{i,t}$ be volume traded by instrument i over bar t . The number of tradable basket units is

$$v_t = \min_i \left\{ \frac{v_{i,t}}{|h_{i,t-1}|} \right\}$$

Transaction costs functions may not be linear, and can be simulated by the strategy.

Method 1.2.14. *ETF Trick: Computation of Allocation Vector with PCA*

Consider an IID multivariate Gaussian process with means vector μ of size $N \times 1$, and covariance matrix V of size $N \times N$. First, perform spectral decomposition $VW = W\Lambda$, where columns in W are reordered so that elements of Λ diagonal are sorted in descending order. Second, given allocations vector ω , portfolio risk is

$$\sigma^2 = \omega' V \omega = \omega' W \Lambda W' \omega = \beta' \Lambda \beta = (\Lambda^{1/2} \beta)' (\Lambda^{1/2} \beta)'$$

where β is projection of ω on orthogonal basis. Third, Λ is a diagonal matrix, thus

$$\sigma^2 = \sum_{n=1}^N \beta_n^2 \Lambda_{n,n}$$

The risk attributed to the n th component is

$$R_n = \beta_n^2 \Lambda_{n,n} \sigma^{-2} = [W' n]_n^2 \Lambda_{n,n} \sigma^{-2}$$

with $R' 1_N = 1$, and 1_N is a vector of N ones.

Note $\{R_n\}_{n=1, \dots, N}$ is distribution of risks across orthogonal components.

Next, compute vector ω which delivers user-defined risk distribution R . Note from earlier,

$$\beta = \left\{ \sigma \sqrt{\frac{R_n}{\Lambda_{n,n}}} \right\}_{n=1, \dots, N}$$

which represents allocation in new (orthogonal basis).

The allocation in old basis is $\omega = W\beta$. Rescaling ω re-scales σ , hence keeping risk distribution constant.

Method 1.2.15. *ETF Trick: Single Futures Roll*

To work with non-negative rolled series, derive price series of \$1 investment as follows:

- i. Compute time series of rolled futures prices
- ii. Compute return r as rolled price change divided by previous roll price
- iii. Form a price series using these returns

These methods allow us to produce a continuous, homogeneous, and structured dataset from collection of unstructured financial data. Note however, that several ML algorithms do not scale well with sample size. ML algorithms achieve higher accuracy when they attempt to learn from relevant examples.

Method 1.2.16. *Sampling for Reduction*

To reduce the amount of data used to fit ML algorithm, downsampling could be used.

- i. Sequential sampling at constant step size (linspace sampling)
- ii. Sampling randomly using uniform distribution (uniform sampling)

Note both samples do not necessarily contain subset of most relevant observations.

Method 1.2.17. *Event-Based Sampling: CUMSUM Filter*

Bets are often placed after some event takes place, hence to let ML algorithm learn whether there is an accurate prediction function under these circumstances, CUSUM filter could be used.

This is a quality-control method, to detect shift in mean value of measured quantity away from a target value. Let $\{y_t\}_{t=1, \dots, T}$ be IID observations arising from a locally stationary process. The cumulative sums are

$$S_t = \max\{0, S_{t-1} + y_t - E_{t-1}[y_t]\}, \quad S_0 = 0$$

An action will be recommended at the first t satisfying $S_t \geq h$ for some threshold h (filter size).

Note $S_t = 0$ whenever $y_t = E_{t-1}[y_t] - S_{t-1}$, The zero floor means some downward deviations will be skipped.

The filter is set up to identify a sequence of upside divergences from any reset level zero.

The threshold is activated when

$$S_t \geq h \Leftrightarrow \exists \tau \in [1, t] \mid \sum_{i=\tau}^t (y_i - E_{i-1}[y_t]) \geq h$$

This concept of run-ups can be extended to include run-downs, giving symmetric CUSUM filter.

$$\begin{aligned} S_t^+ &= \max\{0, S_{t-1}^+ + y_t - E_{t-1}[y_t]\}, \quad S_0^+ = 0 \\ S_t^- &= \min\{0, S_{t-1}^- + y_t - E_{t-1}[y_t]\}, \quad S_0^- = 0 \\ S_t &= \max\{S_t^+, -S_t^-\} \end{aligned}$$

1.2.3 Data Labelling Techniques

Method 1.2.18. *Labelling with Fixed-Time Horizon Method*

Given features matrix X with I rows, $\{X_i\}_{i=1, \dots, I}$ drawn from some bars with index $t = 1, \dots, T$, where $I \leq T$, let an observation X_i be assigned a label $y_i \in \{-1, 0, 1\}$,

$$y_i = \begin{cases} -1 & \text{if } r_{t_{i,0}, t_{i,0}+h} < -\tau \\ 0 & \text{if } |r_{t_{i,0}, t_{i,0}+h}| \leq \tau \\ 1 & \text{if } r_{t_{i,0}, t_{i,0}+h} > \tau \end{cases}$$

$$r_{t_{i,0}, t_{i,0}+h} = \frac{p_{t_{i,0}+h}}{p_{t_{i,0}}} - 1$$

where τ is a pre-defined constant threshold, $t_{i,0}$ is index of bar immediately after X_i takes place, $t_{i,0} + h$ is index of h -th bar after $t_{i,0}$, and $r_{t_{i,0}, t_{i,0}+h}$ is price return over bar horizon h .

Remark 1.2.19. *Limitations of Fixed-Time Horizon Method*

- i. Time bars do not exhibit good statistical properties (as seen earlier)
- ii. The same threshold τ is applied regardless of observed volatility.
Compute daily volatility at intraday estimation points, applying span of n days to an exponentially weighted moving standard deviation.

Method 1.2.20. *Labelling with Triple-Barrier Method*

Labels an observation according to first barrier touched out of three barriers.

- i. Set two horizontal barriers and one vertical barrier. Horizontal barriers are defined by profit-taking and stop-loss limits, which are a dynamic function of estimated volatility (realised or implied). Third barrier is the number of bars elapsed since the position was taken (expiration limit).
- ii. If upper barrier is touched first, label observation as 1. If lower barrier is touched first, label observation as -1 . If vertical barrier is touched first, either label by sign of the return or with 0.

Note that the method is path-dependent. To label an observation, need to account for entire path spanning $[t_{i,0}, t_{i,0} + h]$ where h defines the vertical barrier (expiration limit). Let $t_{i,1}$ be the time of first barrier touch with return as $r_{t_{i,0}, t_{i,1}}$. The horizontal barriers may not be symmetric.

Remark 1.2.21. *Triple-Barrier Method Configurations*

Denote a barrier configuration by triplet $[pt, sl, t1]$ which are the upper barrier, lower barrier, physical barrier. Set value as 0 if barrier is inactive, and 1 if barrier is active.

The three useful configurations are:

- i. $[1, 1, 1]$: to realise profit, but have set a maximum tolerance for losses and a holding period.
- ii. $[0, 1, 1]$: to exit after a number of bars, unless stopped-out.
- iii. $[1, 1, 0]$: take profit as long as not stopped-out.

The three less realistic configurations are:

- i. $[0, 0, 1]$: equivalent to fixed-time horizon method.
- ii. $[1, 0, 1]$: position held until a profit is made or maximum holding period is exceeded, without regard for immediate unrealised losses
- iii. $[1, 0, 0]$: position is held until a profit is made. Could lock in loose position for years.

The two illogical configurations are:

- i. $[0, 1, 0]$: aimless. Hold position until stopped-out.
- ii. $[0, 0, 0]$: no barriers. Position locked forever, no label generated.

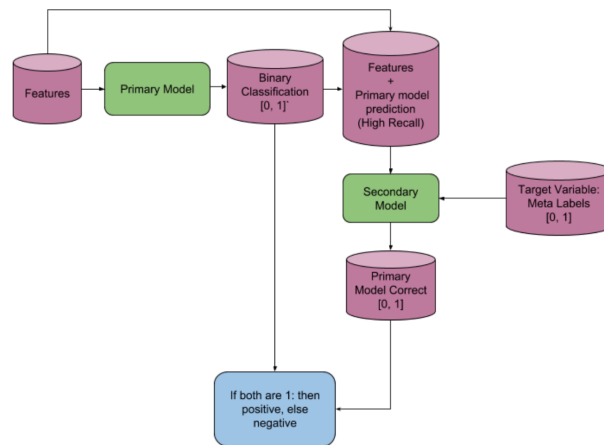


Figure 3: Meta-Labeling Process

Method 1.2.22. *Meta-Labeling*

The technique is particularly helpful to achieve higher F1-scores.

First, build a model that achieves high recall, even if precision is not particularly high. Second, correct for low precision by applying meta-labelling to positives predicted by primary model.

Meta-labelling will filter out false positives, where majority of positives have been identified by primary model. The second model's purpose is to determine if the positive from primary model is true or false.

- i. Train a primary model (binary classification)
- ii. A threshold level is determined at which the primary model has a high recall, ROC curves could be used to help determine a good level.
- iii. Typical features of second model are as follows:
 - i. Primary model features concatenated with predictions from first model.
 - ii. Market state
 - iii. Features indicative of false positives
 - iv. Distribution related
 - v. Recent model performance

Meta Labels are used as target variable in second model. Fit the second model

- iv. Prediction from the secondary model is combined with the prediction from the primary model and only where both are true, is your final prediction true.

Remark 1.2.23. *Limitations of Meta-Labeling*

- i. If model has overfit the data, meta-labelling will not add much value
- ii. If every trade is not treated as an independent observation, the meta-model is forced to determine day-to-day exposures, which is the wrong way to apply the technique
- iii. Technique trades recall for precision. Require a large number of trades to train on, while being happy with reduction in trade frequency

1.2.4 Data Sample Weights

Note that most of ML literature is based on IID assumption, and ML applications usually fail in finance as these assumptions are unrealistic in the case of financial time series.

Remark 1.2.24. *Overlapping Outcomes*

Let label y_i be assigned to an observed feature X_i , where $y_i = f([t_{i,0}, t_{i,1}])$ is a function over the interval. When $t_{i,1} > t_{j,0}$ and $i < j$, then y_j will depend on common return $r_{t_{j,0}, \min\{t_{i,1}, t_{j,1}\}}$ (over interval $[t_{j,0}, \min\{t_{i,1}, t_{j,1}\}]$). The series of labels $\{y_i\}_{i=1, \dots, J}$ are not IID whenever there is overlap between any two consecutive outcomes, i.e., $\exists i \mid t_{i,1} > t_{i+1,0}$. If this is resolved by restricting bet horizon to $t_{i,1} \leq t_{i+1,0}$, there is no overlap, but this will lead to coarse models where features sampling frequency is limited by horizon used to determine outcome. To investigate outcomes that lasted a different duration, samples have to be resampled with different frequency. In addition, if path-dependent labelling technique is to be applied, the sampling frequency will be subordinated to first barrier's touch. Hence, to use $t_{i,1} > t_{i+1,0}$, leading to overlapping outcomes.

Method 1.2.25. *Estimating Uniqueness of Label*

Let two labels y_i and y_j be concurrent at time t , both a function of at least one common return $r_{t-1,t} = \frac{p_t}{p_{t-1}} = 1$. To compute the number of labels that are a function of given return $r_{t-1,t}$:

- i. For each $t = 1, \dots, T$, form a binary array $\{1_{t,i}\}_{i=1, \dots, I}$ where $1_{t,i} \in \{0, 1\}$.
Variable $1_{t,i} = 1$ if and only if $[t_{i,0}, t_{i,1}]$ overlaps with $[t-1, t]$ and $1_{t,i} = 0$ otherwise.
- ii. Compute the number of labels concurrent at t , $c_t = \sum_{i=1}^I 1_{t,i}$

Method 1.2.26. *Average Uniqueness of Label*

To estimate label's uniqueness (non-overlap) across its lifespan.

- i. Uniqueness of label i at time t is $u_{t,i} = 1_{t,i} c_t^{-1}$.
- ii. Average uniqueness of label i is average $u_{t,i}$ over label's lifespan, $\bar{u}_i = (\sum_{t=1}^T u_{t,i})(\sum_{t=1}^T 1_{t,i})^{-1}$.

Note that $\{\bar{u}_i\}_{i=1, \dots, I}$ are not used for forecasting the label, hence there is no information leakage.

Remark 1.2.27. *IID and Oversampling*

Probability of not selecting item i after I draws with replacement on set of I items is $(1 - I^{-1})^I$. As $I \rightarrow \infty$, note that $(1 - I^{-1})^I \rightarrow e^{-1}$. Number of unique observations drawn to be expected is $(1 - e^{-1}) \approx \frac{2}{3}$. If maximum number of overlapping outcomes is $K \leq I$, probability of not selecting a particular item i after I draws with replacement on set of I items is $(1 - K^{-1})^I$. As sample size increase, probability can be approximated as $(1 - I^{-1})^{I \frac{K}{I}} \approx e^{-\frac{K}{I}}$. Implication is that incorrectly assuming IID draws lead to oversampling.

Method 1.2.28. *Sampling with Bootstrap, Redundancy*

Sampling with bootstrapping on observations where $I^{-1} \sum_{i=1}^I \bar{u}_i \ll 1$, in-bag observations will increasingly be redundant to each other, and very similar to out-of-bag observations. Two solutions may be:

- i. Drop overlapping outcomes before performing bootstrap.
As overlaps are not perfect, dropping an observation due to overlap will lead to extreme loss in information.
- ii. Utilise the average uniqueness $I^{-1} \sum_{i=1}^I \bar{u}_i$ to reduce undue influence of outcomes that contain redundant information. Ensure in-bag observations are not sampled at frequency much higher than uniqueness.

Method 1.2.29. *Sequential Bootstrap*

Draws made according to changing probability that controls for redundancy.

- i. Observation X_i is drawn from uniform distribution, $i \sim U[1, I]$.
Probability of drawing any value i is $\delta_i^{(1)} = I^{-1}$.
- ii. Second draw, to reduce probability of drawing observation X_j with highly overlapping outcome.
Let φ be sequence of draws (may include repetitions), where $\{\varphi^{(1)}\} = \{i\}$.
Uniqueness of j at time t is $u_{t,j}^{(2)} = 1_{t,j}(1 + \sum_{k \in \varphi^{(1)}} 1_{t,k})^{-1}$, which is the uniqueness from adding alternative j 's to existing sequence of draws $\varphi^{(1)}$.
Average uniqueness of j is average $u_{t,j}^{(2)}$ over j 's lifespan, $\bar{u}_j^{(2)} = (\sum_{t=1}^T u_{t,j}) (\sum_{t=1}^T 1_{t,j})^{-1}$.
A second draw can be made based on updated probabilities $\{\delta_j^{(2)}\}_{j=1, \dots, I}$:

$$\delta_j^{(2)} = \bar{u}_j^{(2)} \left(\sum_{k=1}^I \bar{u}_k^{(2)} \right)^{-1}$$

where $\sum_{j=1}^I \delta_j^{(2)} = 1$. Do a second draw, update $\varphi^{(2)}$, and re-evaluate $\{\delta_j^{(3)}\}_{j=1, \dots, I}$.

- iii. Process is repeated until I draws have taken place.

Process draws samples much close to IID, verified by increase in $I^{-1} \sum_{i=1}^I \bar{u}_i$.

Method 1.2.30. *Weighting Observations by Uniqueness and Absolute Return*

Let labels be a function for return sign ($\{-1, 1\}$ for standard label, $\{0, 1\}$ for meta-label). The sample weights can be defined in terms of sum of attributed returns over event's life-span, $[t_{i,0}, t_{i,1}]$,

$$\tilde{w}_i = \left| \sum_{t=t_{i,0}}^{t_{i,1}} \frac{r_{t-1,t}}{c_t} \right|, \quad w_t = \tilde{w}_i \left(\sum_{j=1}^I \tilde{w}_j \right)^{-1}$$

where $\sum_{i=1}^I w_i = I$. The method weigh an observation as a function of absolute log returns that can be attributed uniquely to it. Lower returns should be assigned higher weights.

Method 1.2.31. *Time Decay Weighting*

To let sample weights decay as new observations arrive.

Let $d[x] \geq 0 \forall x \in [0, \sum_{i=0}^I \bar{u}_i]$ be time-decay factors multiplying sample weights from earlier.

The final weight has no decay, $d[\sum_{i=1}^I \bar{u}_i] = 1$, and all other weights will adjust relative to that.

Let $c \in (-1, 1]$ be user-defined parameters that determines decay function as follows:

- i. If $c \in [0, 1]$, then $d[1] = c$ with linear decay
- ii. If $c \in (-1, 0)$, then $d[-c \sum_{i=1}^I \bar{u}_i] = 0$, with linear decay between $[-c \sum_{i=1}^I \bar{u}_i, \sum_{i=1}^I \bar{u}_i]$, and $d[x] \forall x \leq -c \sum_{i=1}^I \bar{u}_i$.

If given linear piecewise function $d = \max\{0, a + bx\}$, requirements are met by following boundary conditions:

- i. $d = a + b \sum_{i=1}^I \bar{u}_i = 1 \Rightarrow a = 1 - b \sum_{i=1}^I \bar{u}_i$
- ii. Contingent on c :
 1. $d = a + b \cdot 0 = c \Rightarrow b = (1 - c)(\sum_{i=1}^I \bar{u}_i)^{-1} \forall c \in [0, 1]$
 2. $d = a - bc \sum_{i=1}^I \bar{u}_i = 0 \Rightarrow b = [(c + 1) \sum_{i=1}^I \bar{u}_i]^{-1} \forall c \in (-1, 0)$

In the implementation, decay takes place according to cumulative uniqueness. Note that

- i. $c = 1$ means there is no time decay

- ii. $0 < c < 1$ means weights decay linearly over time, but every observation still receives strictly positive weight, regardless of age
- iii. $c = 0$ means weights converge linearly to zero over time
- iv. $c < 0$ means oldest portion cT of observations receive zero weight (erased from memory)

Method 1.2.32. *Class Weighting*

Weights for underrepresented labels. Critical in classification problems where the most important classes have rare occurrences. To assign higher weights to samples associated with those rare labels.

1.2.5 Fractionally Differentiated Features

Standard stationarity transformations (i.e. integer differentiation) reduce signal by removing memory. Although stationarity is necessary for inferential purposes, it is rarely the case that we want all memory to be erased. Fractionally differentiated processes exhibit long-term persistence and anti-persistence, hence enhancing the forecasting power compared to standard ARIMA approach.

Definition 1.2.33. *BackShift Operator*

Let B be the backshift operator applied to a matrix of real-valued features $\{X_t\}$, where $B^k X_t = X_{t-k}$ for any integer $k \geq 0$. By binomial expansion, we then have

$$\begin{aligned} (1 - B)^d &= \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k = \sum_{k=0}^{\infty} \prod_{i=0}^{k-1} (d - i) \frac{(-B)^k}{k!} \\ &= \sum_{k=0}^{\infty} (-B)^k \prod_{i=0}^{k-1} \frac{d - i}{k - i} \\ &= 1 - dB + \frac{d(d-1)}{2!} B^2 - \frac{d(d-1)(d-2)}{3!} B^3 + \dots \end{aligned}$$

Remark 1.2.34. *Properties of Fractionally Differentiated Features*

Let d be a real (non-integer) positive number. The arithmetic series consists of dot product

$$\begin{aligned} \tilde{X}_t &= \sum_{k=0}^{\infty} \omega_k X_{t-k} \\ \omega &= \left\{ 1, -d, \frac{d(d-1)}{2!}, -\frac{d(d-1)(d-2)}{3!}, \dots, (-1)^k \prod_{i=0}^{k-1} \frac{d-i}{k-i}, \dots \right\} \\ X &= \{X_t, X_{t-1}, \dots, X_{t-k}, \dots\} \end{aligned}$$

where ω are the weights, X are the values. Properties of these features are:

- i. Long memory: if d is a positive integer number, then

$$\prod_{i=0}^{k-1} \frac{d-i}{k-i} = 0 \quad \forall k > d$$

and memory beyond that point is cancelled.

- ii. Iterative weight generation: given sequence of weights ω , for $k = 0, \dots, \infty$, the weights are

$$\omega_k = -\omega_{k-1} \frac{d - k + 1}{k}, \quad \omega_0 = 1$$

- iii. Convergence: For $k > d$, if $\omega_{k-1} \neq 0$, then

$$\left| \frac{\omega_k}{\omega_{k-1}} \right| = \left| \frac{d - k + 1}{k} \right| < 1$$

and $\omega_k = 0$ otherwise. Hence weights converge asymptotically to zero.

For positive d and $k < d + 1$, then $\frac{d-k+1}{k} \geq 0$, which makes initial weights alternate in sign.

For non-integer d , once $k \geq d + 1$, ω_k will be negative if $\text{int}[d]$ is even, and positive otherwise.

In summary, $\lim_{k \rightarrow \infty} \omega_k = 0^-$ when $\text{int}[d]$ is even, and $\lim_{k \rightarrow \infty} \omega_k = 0^+$ when $\text{int}[d]$ is odd.

In special case $d \in (0, 1)$, that $-1 < \omega_k < 0 \quad \forall k > 0$. Alternate weight signs makes $\{\tilde{X}_t\}_{t=1, \dots, T}$ stationary, as memory wanes or is offset over the long run.

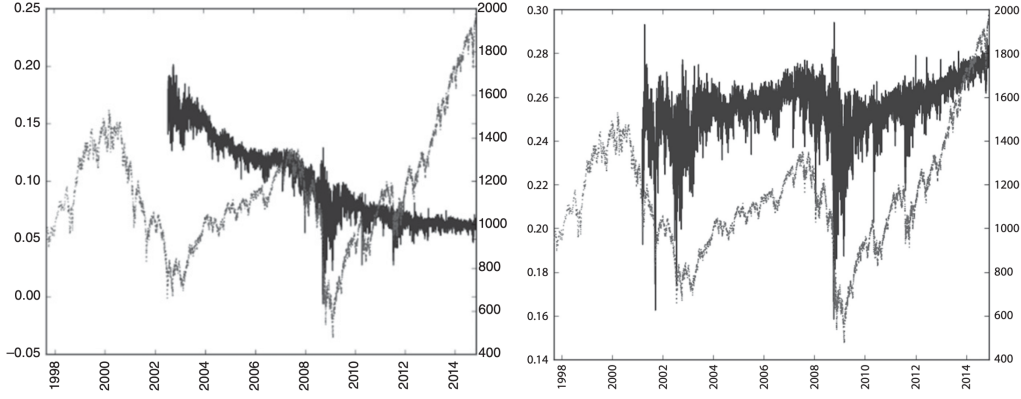


Figure 4: Fractional differentiation controlling for weight loss with expanding and fixed-width window

Method 1.2.35. *Expanding Window*

Given time series T with real observations $\{X_t\}_{t=1,\dots,T}$, for each l , the relative weight loss is defined as

$$\lambda_l = \frac{\sum_{j=T-l}^T |\omega_j|}{\sum_{i=0}^{T-1} |\omega_i|}$$

Given tolerance level $\tau \in [0, 1]$, determine value l^* such that $\lambda_{l^*} \leq \tau$ and $\lambda_{l^*+1} > \tau$. This value l^* corresponds to the first results $\{\tilde{X}_t\}_{t=1,\dots,l^*}$, where weight-loss is beyond acceptable threshold $\lambda_t > \tau$.

From Remark 1.2.34, it is clear λ_{l^*} depends on convergence speed of $\{\omega_k\}$, which in turn depends on $d \in [0, 1]$. For $d = 1, \omega_k = 0 \forall k > 1$, and $\lambda_l = 0 \forall l > 1$, hence it suffices to drop \tilde{X}_1 .

As $d \rightarrow 0^+$, l^* increases, and larger portion of initial $\{\tilde{X}_t\}_{t=1,\dots,l^*}$ needs to be dropped to keep the weight loss $\lambda_{l^*} < \tau$. Note that there will be negative drift caused by negative weights added to initial observations as window is expanded. By controlling for weight loss, negative drift is still substantial as $\{\tilde{X}_t\}_{t=l^*+1,\dots,T}$ are computed on an expanding window.

Method 1.2.36. *Fixed-Width Window*

Drop weights after their modulus $|\omega_k|$ decreases below a given threshold τ . This is equivalent to finding the first l^* such that $|\omega_{l^*}| \geq \tau$ and $|\omega_{l^*+1}| \leq \tau$, setting a new variable $\tilde{\omega}_k$:

$$\tilde{\omega}_k = \begin{cases} \omega_k & \text{if } k \leq l^* \\ 0 & \text{if } k > l^* \end{cases}, \quad \tilde{X}_t = \sum_{k=0}^{l^*} \tilde{\omega}_k X_{t-k} \text{ for } t = T - l^* + 1, \dots, T$$

Note that the same vector of weights is used across all estimates of $\{\tilde{X}_t\}_{t=l^*,\dots,T}$, hence avoiding negative drift caused by expanding window's added weights.

Distribution has skewness and excess kurtosis from memory, but it is stationary.

2 Mathematical Primer

Quantitative trading requires mastery in the following fields of math:

- i. Time Series Analysis
- ii. Stochastic Processes
- iii. Machine Learning

This section serves as a guide in providing the fundamental knowledge required.

2.1 Time Series Analysis

Based on the books by James Douglas [Hamilton \(1994\)](#), and ...

2.1.1 Stationary Time Series

Definition 2.1.1. A *linear first-order difference equation* is defined as

$$y_t = \phi y_{t-1} + w_t$$

where y_t is the target variable (with y_{t-1} the lag 1 variable), w_t is an input variable at time t

The difference equation may be solved by recursive substitution to arrive at

$$y_t = \phi^{t+1}y_{-1} + \phi^t w_0 + \phi^{t-1}w_1 + \phi^{t-2}w_2 + \cdots + \phi w_{t-1} + w_t$$

The *dynamic multiplier* calculates effect of w_t on y_{t+j} , and is given by

$$\frac{\partial y_{t+j}}{\partial w_t} = \phi^j$$

If $|\phi| < 1$, the system is stable. If $|\phi| > 1$, then the system is explosive.

We may generalise the process to *p-th order difference equation*, i.e.,

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + w_t$$

We may rewrite this as a first-order difference equation in a $(p \times 1)$ vector ξ_t :

$$\xi_t = \begin{bmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+1} \end{bmatrix}$$

Define the $(p \times p)$ matrix \mathbf{F} by

$$\mathbf{F} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \cdots & \phi_{p-1} & \phi_p \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Finally, define the $(p \times 1)$ vector \mathbf{v}_t by

$$\mathbf{v}_t = \begin{bmatrix} w_t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Then the system of p equations $\xi_t = \mathbf{F}\xi_{t-1} + \mathbf{v}_t$ is identical to the first order difference equation, and by recursive substitution, we have the following for the case of p -th order difference equation:

$$\xi_{t+j} = \mathbf{F}^{j+1}\xi_{t-1} + \mathbf{F}^j \mathbf{v}_t + \mathbf{F}^{j-1}\mathbf{v}_{t+1} + \cdots + \mathbf{F}\mathbf{v}_{t+j-1} + \mathbf{v}_{t+j}$$

Hence the dynamic multiplier is then $\frac{\partial y_{t+j}}{\partial w_t} f_{11}^{(j)}$, where $f_{11}^{(j)}$ is the $(1, 1)$ element of \mathbf{F}^j .

2.1.2 Univariate Time Series Models

Trading activities through an exchange can be described by a sequence of time stamps ('ticks') $t_0 < t_1 < \dots < t_n$, and 'marks' y_i at time t_i , where t_i denote market open, t_n denote market close. The marks y_i is a characteristic of the order book at time of i th activity. Events with marks associated with the ticks can be described mathematically as a marked point process.

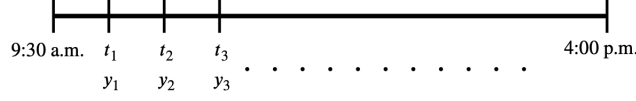


Figure 5: Ticks and Marks

A data aggregation method is where aggregation is conducted when there is a change in marker. An alternative aggregation method would be to divide the time span T for exchange hours into K intervals, so regularly spaced intervals are of size $\Delta t = T/K$. The time series method to be discussed will apply to all aggregated data.

Let $p_{it} = \ln P_{it}$ denote price of i th asset at time t ; $p_t = (p_{1t}, p_{2t}, \dots, p_{nt})$ denote price vector for n assets; y_{it} denote vector of characteristics of i th asset at time t . These quantities are aggregated from high frequency data. Consider r factors $f_t = (f_{1t}, f_{2t}, \dots, f_{rt})$ that may include market and industry factors, and asset characteristics. Trading rules can be broadly grouped as follows:

- i. **Statistical Arbitrage**: $E(p_{i,t+1} \mid p_{i,t}, p_{i,t-1}, \dots, y_{i,t}, y_{i,t-1}, \dots)$, that predicts the price of i th stock at $t+1$ based on past trading information (also known as time series momentum)
- ii. **Momentum**: $E(p_{t+1} \mid p_t, p_{t-1}, \dots, y_t, y_{t-1}, \dots)$, that predicts the cross-sectional momentum of a subset of stocks based on past trading characteristics. For portfolio formation and rebalancing, pairs trading
- iii. **Fair Value**: $E(p_{t+1} \mid p_t, p_{t-1}, \dots, y_t, y_{t-1}, \dots, f_t, f_{t-1}, \dots)$, predicts price using all relevant quantities. Factors normally include market, Fama-French; at a more macro level than timescale considered for price prediction, but may still be useful.

Hence price and volatility prediction can be formulated as a time series prediction problem. Autocorrelations and partial autocorrelations can be used to build autoregressive and ARCH models with some predictive power.

Let Y_1, Y_2, \dots, Y_T be a sequence of random variables with a joint probability distribution. A sequence of observations of stochastic process $\{Y_t, t = 1, \dots, T\}$ is a realisation of the process.

A time series $\{Y_t\}$ is **stationary** if for every integer m , the set of variables $Y_{t_1}, Y_{t_2}, \dots, Y_{t_m}$ depends only on the distance between times t_1, t_2, \dots, t_m . Thus $E(Y_t) = \mu$, $Var(Y_t) = \sigma^2$ are constant for all t .

The **auto-covariance** function is defined as

$$\gamma(s) = Cov(Y_t, Y_{t-s}) = E[(Y_t - \mu)(Y_{t-s} - \mu)] \quad \forall s = 0, \pm 1, \dots$$

The **auto-correlation** of process at lag s is defined as

$$\rho(s) = Corr(Y_t, Y_{t-s}) = \frac{Cov(Y_t, Y_{t-s})}{(Var(Y_t) Var(Y_{t-s}))^{1/2}} = \frac{\gamma(s)}{\gamma(0)}, \quad \forall s = 0, \pm 1, \dots$$

Some examples of stationary stochastic processes are as follows:

Example 2.1.2. (White Noise) Sequence of discrete independent random variables (r.v.) $\{\epsilon_t\}$ with $E[\epsilon_t] = 0$ and $E[\epsilon_t^2] = \sigma^2$. Set $Y_t = \mu + \epsilon_t$, then $E[Y_t] = \mu$, $Cov(Y_t, Y_{t-s}) = Var(Y_t) = \sigma^2$ if $s = 0$, and $Cov(Y_t, Y_{t-s}) = 0$ if $s \neq 0$. Hence the process is stationary.

Example 2.1.3. (Moving Average) Let $\{\epsilon_t\}$ be independent r.v., with process $\{Y_t\}$ where $Y_t = \mu + \epsilon_t + \epsilon_{t-1}$ for $t = 0, 1, 2, \dots$, and μ is constant. Then $E[Y_t] = \mu \forall t$, and

$$Cov(Y_t, Y_{t-s}) = \gamma(s) = \begin{cases} 2\sigma^2 & \text{if } s = 0 \\ \sigma^2 & \text{if } s = 1 \\ 0 & \text{if } s > 1 \end{cases}$$

Hence $\{Y_t\}$ is stationary, with $\rho(s) = \gamma(s)/\gamma(0)$ such that $\rho(0) = 1$, $\rho(1) = 1/2$, $\rho(s) = 0$ for $|s| > 1$.

Some examples of non-stationary stochastic processes are as follows:

Example 2.1.4. (*Random Walk with Drift*) Let $\{\epsilon_t\}$ be sequence of independent r.v. with $E[\epsilon_t] = 0$, $E[\epsilon_t^2] = \sigma^2$, and define process $\{Y_t\}$ by $Y_t = Y_{t-1} + \delta + \epsilon_t$ with $Y_0 = 0$. Then the process can be summarised as

$$Y_t = \delta t + \sum_{j=1}^t \epsilon_j$$

Note that $E[Y_t] = \delta t$ and $Var[Y_t] = t\sigma^2$, hence process $\{Y_t\}$ is not stationary.

Given T observations from stationary process $\{Y_t\}$, the *sample mean* is $\bar{Y} = \frac{1}{T} \sum_{t=1}^T Y_t$. The *sample auto-covariance* function is defined by $\hat{\gamma}(s) = \frac{1}{T} \sum_{t=1}^{T-s} (Y_t - \bar{Y})(Y_{T-s} - \bar{Y})$ for $s = 0, 1, \dots$. The *sample auto-correlation* function (ACF) is defined as $\hat{\rho}(s) = \frac{\hat{\gamma}(s)}{\hat{\gamma}(0)} = r(s)$. The *sample variance* is $Var(\bar{Y}) = \frac{\gamma(0)}{T} \left[1 + 2 \sum_{u=1}^{Y-1} \left(\frac{T-u}{T} \right) \rho(u) \right]$; note that it has to account for auto-correlations.

Definition 2.1.5. A stochastic process $\{Y_t\}$ is a *linear process* if it can be represented as

$$Y_t = \mu + \sum_{j=0}^{\infty} \Psi_j \epsilon_{t-j}$$

where ϵ_t are independent with mean 0, variance σ_ϵ^2 , and $\sum_{j=0}^{\infty} < \infty$.

2.2 Classical Machine Learning

2.2.1 Ensemble Methods

2.3 Deep Learning

2.3.1 Deep Feedforward Networks

Remark 2.3.1. *Deep Feedforward Networks (DFNs)*

Defines a mapping $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ and learns value of parameters $\boldsymbol{\theta}$ that results in best function approximation.

Remark 2.3.2. *Linear Models*

Linear models may fit data efficiently and reliably, either in closed form or with convex optimisation. Model capacity limited to linear functions, does not model interaction between any two input variables

Remark 2.3.3. *Nonlinear Models*

To represent nonlinear functions of \mathbf{x} , apply linear model to transformed input $\phi(\mathbf{x})$ where ϕ is a nonlinear transformation. Kernel trick may be applied to obtain nonlinear learning algorithm. To choose mapping ϕ :

- i. Choose generic ϕ , such as that used by kernel machines based on RBF kernel.
If $\phi(\mathbf{x})$ is of high enough dimension, can find enough capacity to fit training set, but generalisations to test set remains poor. Mappings are based on principle of local smoothness and do not encode enough prior information to solve advanced problems.
- ii. Manually engineer ϕ . Requires decades of human effort for each task, and practitioners specialising in different domains (i.e, speech recognition, computer vision) with little transfer between domains.
- iii. Learn ϕ through deep learning. Model is $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}, \mathbf{w}) = \phi(\mathbf{x}; \boldsymbol{\theta})^T \mathbf{w}$, where parameters $\boldsymbol{\theta}$ can be used to learn ϕ from broad class of functions, and parameters \mathbf{w} that map from $\phi(\mathbf{x})$ to desired output.
Do not require training problem to be convex, and only require finding the right general function.

Definition 2.3.4. *Cost Functions*

- i. Learning Conditional Distributions with Maximum Likelihood: cost function is negative log-likelihood, which is the cross-entropy between training data and model distribution:

$$J(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{y} \mid \mathbf{x})$$

Specific form of cost function changes from model to model, depending on form of $\log p_{\text{model}}$.

Method removes the burden of designing cost functions for each model, as specifying a model $p(\mathbf{y} \mid \mathbf{x})$ automatically determines a cost function $\log p(\mathbf{y} \mid \mathbf{x})$.

- ii. Learning Conditional Statistics: to learn just one conditional statistic of \mathbf{y} given \mathbf{x} .
With sufficiently powerful neural network, this can represent any function f from a wide class of function, limited by features of continuity and boundedness. Hence the cost function is a functional. Learning is choosing a functional rather than a set of parameters.
By calculus of variable, solving the optimisation problem yields the below function,

$$\begin{aligned} f^* &= \arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p_{\text{data}}} \|\mathbf{y} - f(\mathbf{x})\|^2 \\ f^* &= \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}(\mathbf{y} \mid \mathbf{x})} [\mathbf{y}] \end{aligned}$$

If infinitely many samples from the true data-generating distribution could be trained, then minimising the mean squared error cost function gives a function that predicts mean of \mathbf{y} for each value of \mathbf{x} .

A second result from calculus of variations is:

$$f^* = \arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim p_{\text{data}}} \|\mathbf{y} - f(\mathbf{x})\|_1$$

which is a function that predicts the median value of \mathbf{y} for each \mathbf{x} . This is the mean absolute error.

Note that mean squared error and mean absolute error often lead to poor results when used with gradient-based optimisation. Output units that saturate may produce very small gradients when combined with these cost functions. Hence the reason that cross-entropy cost function is more popular.

Definition 2.3.5. *Output Units*

- i. Linear Units: base on affine transformation with no nonlinearity.
Given features \mathbf{h} , a layer of output units produces vector $\hat{\mathbf{y}} = \mathbf{W}^T \mathbf{h} + \mathbf{b}$. Linear output layers are used to produce mean of conditional Gaussian distribution:

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y}; \hat{\mathbf{y}}, \mathbf{I})$$

Maximising log-likelihood is equivalent to minimising mean squared error.

Linear units do not saturate, hence may be used for wide variety of optimisation algorithms.

- ii. Sigmoid Units: define Bernoulli distribution y conditioned on \mathbf{x} . Neural net to predict $P(y = 1 \mid \mathbf{x})$, which lies in interval $[0, 1]$. The sigmoid unit is defined by

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{h} + b)$$

where σ is the logistic sigmoid function.

Note that the cost function used with maximum likelihood is $-\log P(y \mid \mathbf{x})$, preventing saturation.

The loss function for MLE of Bernoulli parametrised by sigmoid is

$$J(\boldsymbol{\theta}) = -\log P(y \mid \mathbf{x}) = -\log \sigma((2y - 1)z) = \zeta((1 - 2y)z)$$

Function saturates only when $(1 - 2y)z$ is very negative, i.e., when model has the right answer.

- iii. Softmax Units: used to represent probability distribution over n different classes.
A linear layer predicts unnormalised log probabilities:

$$\mathbf{z} = \mathbf{W}^T \mathbf{h} + \mathbf{b}$$

where $z_i = \log \tilde{P}(y = i \mid \mathbf{x})$. Softmax function is then

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

The function is to maximise $\log P(y = i; \mathbf{z}) = \log \text{softmax}(\mathbf{z})_i = z_i - \log \sum_j \exp(z_j)$.

Note that the input z_i always has direct contribution to cost function, and the term cannot saturate.

Un-regularised maximum likelihood will drive the model to learn parameters that drive the softmax to predict fraction of counts for each outcome observed in the training set:

$$\text{softmax}(\mathbf{z}(\mathbf{x}; \boldsymbol{\theta}))_i \approx \frac{\sum_{j=1}^m \mathbf{1}_{y^{(j)}=i, \mathbf{x}^{(j)}=\mathbf{x}}}{\sum_{j=1}^m \mathbf{1}_{\mathbf{x}^{(j)}=\mathbf{x}}}$$

Note that objective functions other than log-likelihood does not work well with softmax function. Squared error is poor loss function for softmax units, and can fail to train the model to change its output.

Softmax function can saturate, and many functions based on softmax also saturate, unless they are able to invert the saturating activating function.

- iv. Other Output Types: generally, given a conditional distribution $p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$, principle of maximum likelihood suggests using $-\log p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta})$ as the cost function.
Neural networks represent a function $f(\mathbf{x}; \boldsymbol{\theta}) = \boldsymbol{\omega}$. The outputs of the function are not direct predictions of value \mathbf{y} , but the parameters for distribution over \mathbf{y} . The loss function is then $-\log p(\mathbf{y}; \boldsymbol{\omega}(\mathbf{x}))$.

Remark 2.3.6. *Learning Distribution Parameters*

- i. Heteroscedastic Model: to predict different variance in \mathbf{y} for different values of \mathbf{x} , formulate the Gaussian distribution using precision rather than variance. In multi-variate case, the diagonal precision matrix is used, $\text{diag}(\boldsymbol{\beta})$. The log-likelihood of Gaussian distribution parametrised by $\boldsymbol{\beta}$ involves only multiplication by β_i and addition of $\log \beta_i$. The gradient of these operations are well-behaved.
Let $\boldsymbol{\alpha}$ be raw activation of the model used to determine diagonal precision. The softplus function may be used to obtain a positive precision vector $\boldsymbol{\beta} = \zeta(\boldsymbol{\alpha})$. Same strategy applies equally if using variance or standard deviation rather than precision.
If covariance is full and conditional, then parametrisation must be chosen that guarantees positive-definiteness of predicted covariance matrix.

$$\boldsymbol{\Sigma}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{B}^T(\mathbf{x})$$

where \mathbf{B} is unconstrained square matrix. Note that if matrix is full rank, then computing likelihood requires $O(d^3)$ a $d \times d$ matrix for the determinant and inverse of $\boldsymbol{\Sigma}(\mathbf{x})$.

- ii. Mixture Density Networks: to perform multimodal regression (predict real values that come from conditional distribution $p(\mathbf{y} \mid \mathbf{x})$ that can have several different peaks in \mathbf{y} for the same \mathbf{x}).

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{i=1}^n p(c = i \mid \mathbf{x}) \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}^{(i)}(\mathbf{x}), \boldsymbol{\Sigma}^{(i)}(\mathbf{x}))$$

The neural network will have three outputs:

1. Mixture components $p(c = i | \mathbf{x})$, forming a multinoulli distribution over n different components with latent variable c , obtained by softmax over n -dimensional vector.
2. Means $\boldsymbol{\mu}^{(i)}(\mathbf{x})$
3. Covariances $\boldsymbol{\Sigma}^{(i)}(\mathbf{x})$

Gradient-based optimisation of conditional Gaussian mixtures can be unreliable as the divisions can be numerically unstable. May be solved by clipping gradients, or scaling gradients heuristically.

Definition 2.3.7. Hidden Units

Even if hidden units are not differentiable at all input points, gradient descent still performs well enough as the training algorithms do not usually arrive at local minimum of cost function, but reduce its value significantly. Most hidden units has input vector \mathbf{x} , computes an affine transformation $\mathbf{z} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$, then applying element-wise non-linear function $g(\mathbf{z})$.

- i. Rectified Linear Units (ReLUs): uses activation function $g(z) = \max\{0, z\}$. Typically used on top of an affine transformation $\mathbf{h} = g(\mathbf{W}^T \mathbf{x} + \mathbf{b})$. In initialisation, set all elements of \mathbf{b} to small positive values, so that ReLUs will be initially active for most inputs in training set. Generalisations of ReLUs have non-zero slope α_i for $z_i < 0$: $h_i = \max(0, z_i) + \alpha_i \min(0, z_i)$.
 1. Absolute Value Rectification: sets $\alpha_i = -1$ to obtain $g(z) = |z|$. Used for object recognition from images, to seek features that are invariant under polarity reversal of input illumination.
 2. Leaky ReLU: fixes α_i to small positive value
 3. Parametric ReU: treats α_i as a learnable parameter
- ii. Maxout Units: divide \mathbf{z} into groups of k values. Each maxout unit then outputs maximum element of one of these groups: $g(\mathbf{z})_i = \max_{j \in \mathbb{G}^{(i)}} z_j$, where $\mathbb{G}^{(i)}$ is indices of inputs for group i , which is $\{(i-1)k+1, \dots, ik\}$. This allows learning of piecewise linear function that responds to multiple directions in input \mathbf{x} space. Maxout units learn the activation function itself. With large k , maxout unit can learn to approximate any convex function. Each maxout unit is parametrised by k weight vectors, hence need more regularisation than ReLUs. Benefits include:
 1. Can work well without regularisation if training set is large and number of pieces per unit is low.
 2. Can gain statistical and computation advantages by requiring fewer parameters.
 3. Have redundancy that resists catastrophic forgetting, where neural networks forgot how to perform tasks that were trained on in the past.
- iii. Logistic Sigmoid and Hyperbolic Tangent: the logistic sigmoid activation function is $g(z) = \sigma(z)$, and the hyperbolic tangent activation function is $g(z) = \tanh(z)$. Note that $\tanh(z) = 2\sigma(2z) - 1$. Sigmoidal units saturate across most of the domain, which makes gradient-based learning very difficult. Hence the use in hidden units in feedforward networks is now discouraged. If sigmoidal activation function must be used, hyperbolic tangent activation function performs better, as it resembles identity function more closely. Training $\hat{y} = \mathbf{w}^T \tanh(\mathbf{U}^T \tanh(\mathbf{V}^T \mathbf{x}))$ resembles training a linear model $\hat{y} = \mathbf{w}^T \mathbf{U}^T \mathbf{V}^T \mathbf{x}$ as long as the activations of the network can be kept small. Sigmoidal functions are more common in recurrent networks, probabilities models, auto-encoders
- iv. Linear Unit: consider neural network layer with n inputs, p outputs, $\mathbf{h} = g(\mathbf{W}^T \mathbf{x} + \mathbf{b})$. Replace with two layers, one using weight matrix \mathbf{U} and the other using weight matrix \mathbf{V} . If first layer has no activation function, then the factored approach is to compute $\mathbf{h} = g(\mathbf{V}^T \mathbf{U}^T \mathbf{x} + \mathbf{b})$. If \mathbf{U} produces q outputs, then \mathbf{U} and \mathbf{V} together only contains $(n+p)q$ parameters, while \mathbf{W} contains np parameters. For small q , this is considerable saving in parameters, while the cost is constraining linear transformation to be low rank. This is an efficient way of reducing number of parameters in the model.
- v. Softmax: naturally represent probability distribution over discrete variable with k possible values. Used only in more advanced architectures that explicitly learn to manipulate memory.
- vi. Radial Basis Function (RBF): function becomes more active as \mathbf{x} approaches a template $\mathbf{W}_{:,i}$. As it saturates to 0 for most \mathbf{x} , it can be difficult to optimise.

$$h_t = \exp\left(-\frac{1}{\sigma_i^2} \|\mathbf{W}_{:,i} - \mathbf{x}\|^2\right)$$

- vii. Softplus: smooth version of rectifier for functional approximation and for conditional distributions of undirected probabilistic models. Usage is generally discouraged.

$$g(a) = \zeta(a) = \log(1 + e^a)$$

viii. Hard tanh: shaped similarly to tanh but bounded.

$$g(a) = \max(-1, \min(1, a))$$

Theorem 2.3.8. *Universal Approximation Theorem*

A feedforward network with linear output layer and at least one hidden layer with any 'squashing' activation function can approximate any Borel measurable function from one finite-dimensional space to another with any desired non-zero amount of error, provided the network is given enough hidden units.

Method 2.3.9. *Architecture Design*

Neural network layers are arranged in a chain structure, with each layer being a function of preceding layer.

$$\begin{aligned} \mathbf{h}^{(1)} &= g^{(1)}(\mathbf{W}^{(1)T} \mathbf{x} + \mathbf{b}^{(1)}) \\ \mathbf{h}^{(k)} &= g^{(k)}(\mathbf{W}^{(k)T} \mathbf{h}^{(k-1)} + \mathbf{b}^{(k)}), \quad k \geq 2 \end{aligned}$$

The main considerations are the depth of network and width of each layer. Deeper networks use far fewer units per layers and far fewer parameters, and often generalise to the test set, but are harder to optimise.

Remark 2.3.10. *Depth of Network and Universal Approximation Theorem*

A feedforward network with single layer is sufficient to represent any function, but the layer may be infeasibly large and fail to learn and generalise correctly. Using deeper models can reduce number of units required to represent the desired function and can reduce generalisation error.

Shallow networks with broad family of non-polynomial activation functions have universal approximation properties. Piecewise linear networks can represent functions with number of regions that is exponential to depth. The number of linear regions carved out by deep rectifier network with d inputs, depth l , and n units per hidden layer is $O(\binom{n}{d}^{d(d-1)} n^d)$. For maxout networks with k filters per unit, this is $O(k^{(k-1)+d})$.

Algorithm Backpropagation learning algorithm

Input:

A set of training examples $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
A multilayer network with L layers, weights w_{ij}^l , and activation function f
Loss function $J(y, o)$
Learning rate $0 < \alpha < 1$
Number of epochs $epochs$

```

1: for each weight  $w_{ij}^l$  in the network do
2:    $w_{ij}^l \leftarrow$  a small random number
3: for  $i = 1$  to  $epochs$  do
4:   for each training example  $(\mathbf{x}, y) \in D$  do
5:     /* Propagate the inputs forward to compute the outputs */
6:     for each neuron  $i$  in the input layer do
7:        $a_i^0 \leftarrow x_i$ 
8:     for  $l = 2$  to  $L$  do
9:       for each neuron  $i$  in layer  $l$  do
10:         $z_i^l \leftarrow \sum_j w_{ij}^l a_j^{l-1}$ 
11:         $a_i^l \leftarrow f(z_i^l)$ 
12:     /* Propagate deltas backward from the output layer to the input layer */
13:     for each neuron  $i$  in the output layer do
14:        $\delta_i^L \leftarrow \frac{\partial J(y_i, o_i)}{\partial o_i} f'(z_i^L)$ 
15:     for  $l = L - 1$  to  $1$  do
16:       for each neuron  $i$  in layer  $l$  do
17:         $\delta_i^l \leftarrow f'(z_i^l) \sum_j (w_{ji}^{l+1} \delta_j^{l+1})$ 
18:     /* Update the weights using the deltas */
19:     for each weight  $w_{ij}^l$  in the network do
20:        $w_{ij}^l \leftarrow w_{ij}^l - \alpha \delta_i^l a_j^{l-1}$ 

```

Figure 6: Backpropagation Algorithm

Remark 2.3.11. *Backpropagation*

To calculate the gradient of loss function with respect to each of individual parameters of the neural network. Model training begins with random initialisation of weights and biases.

- i. Forward pass: input is sampled from training data. Nodes receive input vector and passes their value (multiplied by random initial weight) to nodes of first hidden layer. The hidden units take weighted sum of these output values as an input to the activation function, whose output is used for next hidden layer.
- ii. Error computation: the final output of the network is compared to the ground truth, difference is calculated for the error value.
- iii. Backwards pass: the error value computed earlier is used to compute the gradient of loss function. The gradient is then propagated back through the network, and the weights are updated according to their contribution to the error. The learning rate determines the size of weight updates.
- iv. Weights update: the weights are updated in opposite direction of the gradient

2.3.2 Regularisation for Deep Learning

Definition 2.3.12. *Regularisation* refers to adding a parameter norm penalty $\Omega(\theta)$ to the objective function J . The regularised objective function is then

$$\tilde{J}(\theta; \mathbf{X}, \mathbf{y}) = J(\theta; \mathbf{X}, \mathbf{y}) + \alpha\Omega(\theta)$$

where $\alpha \in [0, \infty)$ is a hyper-parameter that weights the contribution of norm penalty term.

For neural networks, the parameter norm penalty Ω is chosen such that it penalises only the weights of the affine transformation at each layer and leaves the biases un-regularised.

Definition 2.3.13. *L^2 /Ridge Regularisation*

The regularisation term added to objective function is

$$\Omega(\theta) = \frac{1}{2} \|\mathbf{w}\|_2^2$$

Remark 2.3.14. *Behaviour of Weight Decay (L^2) Regularisation*

Assuming no bias parameter, a model have the following objective function and parameter gradient:

$$\begin{aligned} \tilde{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) &= \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + J(\mathbf{w}; \mathbf{X}, \mathbf{y}) \\ \nabla_{\mathbf{w}} \tilde{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) &= \alpha \mathbf{w} + \nabla_{\mathbf{w}} J(\mathbf{w}; \mathbf{X}, \mathbf{y}) \end{aligned}$$

On a single gradient step, the update is as follows:

$$\mathbf{w} \leftarrow (1 - \epsilon\alpha) \mathbf{w} - \epsilon \nabla_{\mathbf{w}} J(\mathbf{w}; \mathbf{X}, \mathbf{y})$$

The addition of weight decay has modified learning rule to multiplicatively shrink weight vector by constant factor on each step before gradient update.

Using quadratic approximation to objective function at minimal unregularised training cost, approximation is

$$\hat{J}(\tilde{\mathbf{w}}) = J(\mathbf{w}^*) + \frac{1}{2} (\mathbf{w} - \mathbf{w}^*)^T \mathbf{H} (\mathbf{w} - \mathbf{w}^*), \quad \mathbf{w}^* = \arg \min_{\mathbf{w}} J(\mathbf{w})$$

where \mathbf{H} is Hessian matrix of J with respect to \mathbf{w} evaluated at \mathbf{w}^* . Minimum of \hat{J} occurs where gradient is

$$\nabla_{\mathbf{w}} \hat{J}(\mathbf{w}) = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) = \mathbf{0}$$

Adding weight decay gradient, solve for minimum of regularised version of \hat{J} . Let $\tilde{\mathbf{w}}$ be minimum, then

$$\begin{aligned} \alpha \tilde{\mathbf{w}} + \mathbf{H}(\tilde{\mathbf{w}} - \mathbf{w}^*) &= \mathbf{0} \\ (\mathbf{H} + \alpha \mathbf{I}) \tilde{\mathbf{w}} &= \mathbf{H} \mathbf{w}^* \\ \tilde{\mathbf{w}} &= (\mathbf{H} + \alpha \mathbf{I})^{-1} \mathbf{H} \mathbf{w}^* \end{aligned}$$

As $\alpha \rightarrow 0$, regularised solution $\tilde{\mathbf{w}} \rightarrow \mathbf{w}^*$. Note \mathbf{H} is real and symmetric, hence decompose into diagonal matrix \mathbf{A} and orthonormal basis of eigenvectors \mathbf{Q} such that $\mathbf{H} = \mathbf{Q} \mathbf{A} \mathbf{Q}^T$, to get

$$\tilde{\mathbf{w}} = (\mathbf{Q} \mathbf{A} \mathbf{Q}^T + \alpha \mathbf{I})^{-1} \mathbf{Q} \mathbf{A} \mathbf{Q}^T \mathbf{w}^* = \mathbf{Q} (\mathbf{A} + \alpha \mathbf{I})^{-1} \mathbf{A} \mathbf{Q}^T \mathbf{w}^*$$

The effect of weight decay rescale \mathbf{w}^* along axes defined by eigenvectors of \mathbf{H} . The component of \mathbf{w}^* aligned with i -th eigenvector of \mathbf{H} is rescaled by factor $\frac{\lambda_i}{\lambda_i + \alpha}$. For components where $\lambda_i \gg \alpha$, the effects of regularisation is relatively small. For components where $\lambda_i \ll \alpha$, components will be shrunk to nearly zero magnitude.

Definition 2.3.15. *L^1 Regularisation*

The L^1 regularisation is the sum of absolute values of individual parameters.

$$\Omega(\boldsymbol{\theta}) = \|\mathbf{w}\|_1 = \sum_i |w_i|$$

Definition 2.3.16. *Behaviour of L^1 Regularisation*

Assuming no bias parameter, a model has the following objective function and parameter gradient:

$$\begin{aligned}\tilde{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) &= \alpha \|\mathbf{w}\|_1 + J(\mathbf{w}; \mathbf{X}, \mathbf{y}) \\ \nabla_{\mathbf{w}} \tilde{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) &= \alpha \text{sign}(\mathbf{w}) + \nabla_{\mathbf{w}} J(\mathbf{w}; \mathbf{X}, \mathbf{y})\end{aligned}$$

Note that the regularisation contribution to gradient is a constant factor with sign equal to $\text{sign}(w_i)$. Minimum of \tilde{J} occurs at where

$$\nabla_{\mathbf{w}} \tilde{J}(\mathbf{w}) = \mathbf{H}(\mathbf{w} - \mathbf{w}^*)$$

Assuming the Hessian is diagonal, $\mathbf{H} = \text{diag}([H_{1,1}, \dots, H_{n,n}])$, where each $H_{i,i} > 0$. The quadratic approximation of regularised objective function is then

$$\begin{aligned}\tilde{J}(\mathbf{w}; \mathbf{X}, \mathbf{y}) &= J(\mathbf{w}^*; \mathbf{X}, \mathbf{y}) + \sum_i \left[\frac{1}{2} H_{i,i} (\mathbf{w}_i - \mathbf{w}_i^*)^2 + \alpha |w_i| \right] \\ w_i &= \text{sign}(w_i^*) \max \left\{ |w_i^*| - \frac{\alpha}{H_{i,i}}, 0 \right\}\end{aligned}$$

For the case where $w_i^* > 0$ for all i , then

- i. if $w_i^* \leq \frac{\alpha}{H_{i,i}}$, the optimal value is $w_i = 0$ as contribution of $J(\mathbf{w}; \mathbf{X}, \mathbf{y})$ to regularised objective $\tilde{J}(\mathbf{w}; \mathbf{X}, \mathbf{y})$ is overwhelmed in direction i by L^1 regularisation which pushes w_i to zero.
- ii. if $w_i^* > \frac{\alpha}{H_{i,i}}$, regularisation shifts optimal value of w_i in the direction by $\frac{\alpha}{H_{i,i}}$.

For the case where $w_i^* < 0$, this happens similarly, but with L^1 penalty decreasing w_i by $\frac{\alpha}{H_{i,i}}$, with min value 0. Note that L^1 produces a more sparse solution, which is a feature selection mechanism.

3 Market Microstructure

3.1 Market Fundamentals

The basic function of a market is to bring buyers and sellers together.

Process 3.1.1. (*Four Components of a Trade*)

- i. Acquisition of information and quotes
 1. Quality information and transparency are crucial to price discovery
 2. Transparent markets quickly disseminate high-quality information
 3. Opaque markets are those that lack transparency
- ii. Routing of the trade order
 1. Selecting the brokers to handle the trades
 2. Deciding which markets will execute the trades and transmitting the trades to the markets
- iii. Execution. Buys are matched and executed against sells according to the rules of the market
- iv. Confirmation, clearance and settlement
 1. Clearance is the recording and comparison of the trade records
 2. Settlement involves the actual delivery of the security and its payment
 3. Might include trade allocation

Remark 3.1.2. (*Risks of Algorithmic Trading*)

- i. Leaks might arise from competitor efforts to reverse engineer them
- ii. Many algorithms lack the capacity to handle or respond to exceptional or rare events.

Remark 3.1.3. (*Cornering-the-Market*) Trader takes huge long futures position and tries to exercise control over supply of underlying commodity. As maturity of futures contract is approaching, position is not closed, number of outstanding contracts exceed commodities available. Holders of short positions desperately try to close positions, leading to rise in both futures and spot prices.

Abuse is dealt with by increasing margin requirements or imposing stricter position limits or prohibiting trades that increase speculator's open position or requiring market participants to close their positions.

3.1.1 Liquidity Access in Equity Markets

(*Exchanges*) Account for 60% to 70% of all activity. The full order-book, arrivals/cancellations are all published, the liquidity information is transparent. Larger orders may impact the market. Liquidity at best price cannot be ignored, as the exchange will need to reroute to other exchanges where price is better while charging a fee (National Best Bid Offer, NBBO). All exchanges have almost exactly the same trading mechanism, and behave exactly the same way during the trading day except for opening and closing auctions.

Most exchanges use maker-taker fee model, but some exchanges (BYX, EDGA, NSX, BX) have an inverted fee model where rebate is provided for taking liquidity. IEX uses a speed bump to remove speed advantages of HFTs, providing a less 'toxic' liquidity pool.

(*Dark Pools/ATS*) Dark pools do not display any order information and use the NBBO as reference price. To avoid accessing protected venues, these pools trade only at the inside market (at or within the bid ask spread). Still possible to identify large blocks of liquidity by 'pinging' the pool at minimum lot size; counteracted by sending orders with minimum fill quantity tag, which allows block to be transparent from small pinging. Most of ATS are run by major investment banks; some venues allow direct trading between investment firms. Note that many of the trading strategies used by firms tend to be highly correlated, hence liquidity is often on the same side; venues have to leverage sell side broker's liquidity to supplement their own.

(*Single Dealer's Platform/Systematic Internalisers*) Broker/Dealer and other institutional clients connect to Single Dealer Platform (SDP) directly. Not regulated ATS, hence can offer unique products. Brokers provide their own SDPs to expose their internal liquidity.

(*Auctions*) Exchanges begin and end day with a primary auction procedure that leverages special order types to accumulate supply and demand, then run algorithm that determines best price that would at best pair off the most volume. An opportunity for active and passive investors to exchange large amounts of liquidity.

3.1.2 Trading Mechanisms

Most common approach by modern electronic exchanges is the time/price priority, *continuous double auction* trading system. There are multiple buyers and multiple sellers participating at the same time.

(*Limit Order Book, LOB*) Stores all non-executed orders with associated instructions. Highly efficient, able to handle a high degree of concurrency to ensure the state is always correct. Has 2 copies of core data structure (for Buy, Sell orders). Supports 3 basic instruction types (insert, cancel, amend). There are 4 events on both Buy and Sell that may alter state of order book (limit order submission, limit order cancellation, limit order amendment, execution).

(*Matching Algorithm*) Responsible for interpreting various events to determine if any buy and sell orders can be matched. When multiple orders can be paired, price/time priority is used, where the order with most competitive prices are matched, and when prices are equal, the order that arrived prior is chosen.

Exchanges will publish order imbalance that exists among orders on opening and closing books during *Open Auctions*, with indicative price and volume. The following are published every second on market data feeds:

- i. *Current Reference Price*: Price at which paired shares are maximised, imbalance is minimised, distance from bid-ask mid-point is minimised, in that order
- ii. *Near Indicative Clearing Price*: The crossing price at which orders in opening, closing book and continuous book would clear against each other
- iii. *Far Indicative Clearing Price*: The crossing price at which orders in opening, closing book would clear against each other
- iv. *Number of Paired Shares*: The number of on-open or on-close shares that is able to pair off at the current reference price
- v. *Imbalance Shares*: The number of opening or closing shares that would remain unexecuted at the current reference price
- vi. *Imbalance Side*: The side of imbalance. B is buy-side imbalance; S is sell-side imbalance; N is no imbalance; O is no marketable on-open or on-close orders

In general, the following rules apply to match supply and demand:

- i. Crossing price must maximise volume transacted
- ii. If several prices result in similar volume transacted, the crossing price is the one the closest from last price
- iii. Cross price is identical for all orders executed
- iv. If two orders are submitted at same price, the order submitted first has priority
- v. It is possible for an order to be partially executed if the other side quantity is not sufficient
- vi. 'At Market' orders are executed against each other at the determined crossing price, up to available matching quantity on both sides, but generally do not participate in price formation process
- vii. For Open Auction, unmatched 'At Market' orders are entered into continuous session of LOB as limit orders at the crossing price

The open auction is a major price discovery mechanism, as it occurs after a period of market inactivity when market participants were unable to transact even if they have information. Market participants with better information are more likely to participate, with more aggressive orders to extract liquidity, hence the mechanism is quite volatile and more suited for short-term alpha investors.

During execution, smaller orders may avoid participation in open auction and period thereafter; larger orders may participate to extract significant liquidity from the market.

Diversity of order types is key component of continuous double auction electronic markets.

Definition 3.1.4.

- i. *Market Order*: trade carried out immediately at best price available in market.
- ii. *Limit Order*: only executed at this price or at one more favourable to the trader.
- iii. *Stop/Stop-Loss Order*: not visible in LOB. Only become active when certain price is reached or passed, then enter order book as either limit or market order depending on user setup.

- iv. *Stop-Limit Order*: combination of stop order and limit order. Order becomes limit order as soon as a bid or ask is made at the price equal to or less favourable than stop price. If stop price and limit price is the same, then the order is *stop-and-limit* order.
- v. *Trailing Stop Order*: function like stop orders, but stop price is set dynamic rather than static.
- vi. *Market-if-Touched (MIT)/Board Order*: executed at best available price after trade occurs at a specified price or more favourable. Ensure profit is taken if sufficiently favourable price movements occur.
- vii. *Market-Not-Held/Discretionary Order*: traded as market order, execution may be delayed at broker's discretion for better price.
- viii. *All-Or-None Order*: request full execution of order. Not executed until full quantity is available.
- ix. *Peg Order*: specify a price level at which order should be continuously and automatically repriced. Used for mid-point executions in non-displayed markets.
- x. *Iceberg Order*: limit order with specific display quantity, designed to prevent information leakage.
- xi. *Hidden Order*: when available to trade, not directly available to other market participants in central LOB
- xii. *On-Open*: request execution at open price. Can be limit-on-open or market-on-open.
- xiii. *On-Close*: request execution at close price. Can be limit-on-close or market-on-close.
- xiv. *Imbalance Only*: provide liquidity intended to offset on-open/on-close order imbalances during opening/-closing cross. These generally are limit orders.
- xv. *D-Quote*: Special order on NYSE, mainly used during close auction period.
- xvi. *Funari*: Special order on Tokyo Stock Exchange, allows limit order placed in book during continuous session to automatically enter closing auction as market orders.

Instruction validity may take the following forms:

- i. *Day Order*: valid for full duration of trading session
- ii. *Extended Day Orders*: allows for trading in extended hours
- iii. *Good-Till-Cancel Order*: in effect until executed or until end of trading in particular contract
- iv. *Immediate-or-Cancel Order*: will be immediately cancelled back to sender after reaching matching engine if it does not get immediate fill.
- v. *Fill-or-Kill Order*: must be executed immediately on receipt or none at all.

3.1.3 Market Microstructure Primer

A market microstructure analysis framework follows three main categories:

- i. Price Formation and Price Discover: how prices impound information over time, and how determinants of trading costs vary
- ii. Market Design: impact of trading rules on price formation. Choice of tick size, circuit breakers that halt trading in event of large price swings, degree of anonymity, transparency of information to market participants. These create a diverse set of constraints and opportunities.
- iii. Transparency: quantity, quality and speed of information effect on trading process. Classified into pre-trade (lit order book), post-trade (trade reporting to public).

Topics for research in microstructure includes the following:

- i. Parent order sliced into several child orders sent to market for execution; difficult to discern the informed trader who use sophisticated dynamic algorithms. Retail trades usually cross the spread.
- ii. Understanding of trading intensity in short intervals. Order imbalance is empirically shown to be unrelated to price levels.
- iii. Informed traders usage of hidden orders in entering and exiting the market require further studies.
- iv. Traders respond to changing market conditions by revising quoted prices. Quote volatility can provide valuable information about perceived uncertainty in the market.

4 Equities Trading

5 Fixed Income Trading

6 Derivatives Trading

Based on the classic by John C. Hull (2021)

6.1 Fundamentals of the Market

The derivatives market is much larger than the stock market in terms of underlying assets. Derivatives may be used for hedging, speculation, or arbitrage; and also transfer a wide range of risks from one entity to another.

Definition 6.1.1. A *derivative* involves two parties agreeing to a future transaction, with value depending on the values of other underlying variables.

A derivatives exchange is a market where individuals and companies trade standardised contracts as defined by the exchange. Once two traders have agreed to trade a product offered by the exchange, it is handled by the exchange clearing house, which takes care of the credit risk by requiring each trade to deposit margin.

If the trade is taken over-the-counter (OTC), participants may present it to a central counterparty (CCP) or clear the trade bilaterally. With the 2008 financial crisis, OTC market is forced to become more like the exchange-traded market, with changes as follows:

- i. Standardised OTC derivatives between two financial institutions in US must, whenever possible, be traded on a swap execution facility (SEF), where market participants can post bid and ask quotes, and can trade by accepting the quotes of other market participants.
- ii. Require that a CCP be used for most standardised derivatives transactions between financial institutions.
- iii. All trades must be reported to a central repository.

6.1.1 Forward, Futures, and Options

Definition 6.1.2. A *spot contract* is an agreement to buy or sell an asset almost immediately. A *forward contract* is an agreement to buy or sell an asset at a certain future time for a certain price.

Let S_T be the spot price of asset at maturity, K is delivery price.

The payoff from a long position in a forward contract is $S_T - K$.

The payoff from a short position in a forward contract is $K - S_T$.

Futures contract are traded on an exchange, unlike forwards which are traded OTC. Majority of futures contract do not lead to delivery, as positions are closed prior to delivery period by entering an opposite trade to the original one.

Party in short position may file notice of intention to deliver with the exchange when they are ready to deliver. If the asset is a commodity, the grade of commodity are specified. The contract size specifies the amount of asset that has to be delivered. The place for delivery must also be specified, as commodities may involve significant transportation costs. The delivery month of the commodity may also be specified, and are chosen by the exchange to meet the needs of market participants. Trading typically ceases a few days before the last day on which delivery can be made. Daily price movement limits are also specified by exchange to prevent speculative excess causing large price movements; in this case trading ceases for the day.

As the delivery price for a futures contract is approached, the futures price converges to the spot price of the underlying asset.

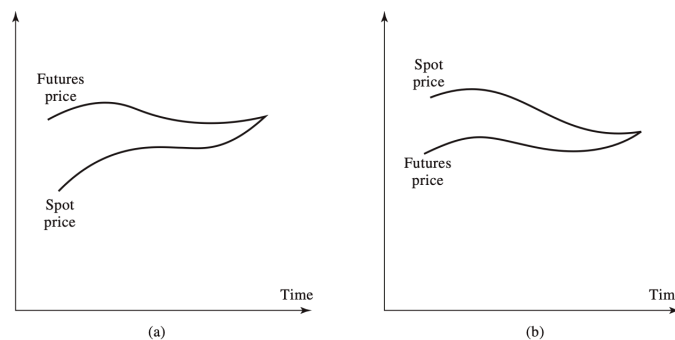


Figure 7: Convergence of futures price to spot price.

Suppose futures price is above spot price during delivery period. Traders have clear arbitrage opportunity: short futures, long asset, make delivery. The futures price will then fall. If futures price is below spot price during delivery period, traders will long futures, wait for delivery, and the futures price will then rise.

Options are traded both on exchanges and OTC.

Definition 6.1.3. A *call option* (*put option*) gives the holder the right to sell (buy) the underlying asset by a certain date for a certain price.

Definition 6.1.4. An *American option* can be exercised at any time up to expiration date. An *European option* can only be exercised on expiration date itself.

6.1.2 Clearing House

Margin accounts are used by exchanges to organise trading so that contract defaults are avoided. Trader has to keep funds in a margin account; the amount to be deposited at the time the contract is entered into is the *initial margin*. At each of trading day, margin account is adjusted to reflect trader's P&L (*daily settlement, marking to market*). Daily settlement leads to funds flowing daily between traders with long positions and traders with short positions; this daily flow of funds to reflect P&L is the *variation margin*. Trading via brokers requires a *maintenance margin*, which is lower than initial margin; if balance in margin account falls below maintenance margin, trader receives *margin call* and is required to top up to initial margin level within short period of time, or else broker closes out the position. Trader is also entitled to withdraw any balance in margin account that is in excess of the initial margin. Brokers pay interest on balance in margin account. A forward contract is settled at end of life, while futures contract is settled daily. Minimum levels for initial and maintenance margins are set by exchange clearing house. Minimum margin levels are determined by variability of the price of underlying asset, revised when necessary.

The clearing house acts as intermediary in futures transactions, and keep track of all daily transactions for calculating net positions of each of its members. Members must provide initial margin reflecting the total number of contracts cleared. The maintenance margin is set equal to initial margin. In determining margin requirements, the number of contracts outstanding is calculated on a net basis rather than a gross basis. Members are required to contribute to a guaranty fund, which is used in the event that a member defaults, and the member's margin is insufficient to cover losses.

6.1.3 OTC Markets

OTC markets use central counterparties (CCPs), which perform the same role as exchange clearing houses. Members of CCP provide both initial margin and daily variation margin, and contribute to a guaranty fund. If an OTC derivative transaction has been agreed upon between parties A and B, and CCP accepts the transaction, they become the counterparty to both A and B. CCP hence takes on credit risk of both A and B. Transactions are valued daily, and there are daily variation margin payments between members.

For bilaterally cleared OTC, two companies enter a master agreement covering all their trades (ISDA). The agreement includes a credit support annex (CSA), requiring both parties to provide collateral. Collateral agreements in CSAs usually require transactions to be valued daily. Since 2016, regulations require both initial margin and variation margin between financial institutions.

Definition 6.1.5. *Credit spread* is the difference between the interest rate and risk free rate.

(*Treasury Rates*) The rates on Treasury bills and Treasury bonds. The Treasury rates of developed countries are regarded as risk-free rates, as it is assumed that there is no chance the government will default.

(*Overnight Rates*) Borrowing and lending overnight by financial institutions to match asset and liabilities requirements for reserves; the overnight rate is the *federal funds rate*. The weighted average of rates in brokered transactions is the *effective federal funds rate*. The Federal Reserve may intervene with its own transactions to raise or lower the rates.

(*Repo Rates*) Secured borrowing rates; the difference between the price at which securities are sold and then repurchased. If structured carefully, involves very little credit risk. Most common type of repo is an *overnight repo*. In longer term arrangements, *term repos* are used.

The terminal value of an investment A invested at interest rate of R per annum, compounded m times per annum, is $A(1 + \frac{R}{m})^{mn}$. If $m = 1$, the rate is the *equivalent annual interest rate*. If continuous compounded is used, then the terminal value at the end of a year is Ae^R .

The *n -year spot-rate* is the interest rate earned on a zero-coupon bond. The *Bond Price* is the present value of all cash flows that will be received by owner of the bond, with different spot rate for each cash flow. The *Bond Yield* is the discount rate that, when applied to all cash flows, gives a bond price equal to its market price. The *Par Yield* for a certain bond maturity is the coupon rate that causes the bond price to equal its par value.

Definition 6.1.6. *Forward Rates* are rates implied by current spot rates for periods of time in the future.

Given R_1, R_2 the spot rates for maturities T_1, T_2 respectively, and R_F the forward rate between T_1 and T_2 , then

$$R_F = \frac{R_2 T_2 - R_1 T_1}{T_2 - T_1} = R_2 + (R_2 - R_1) \frac{T_1}{T_2 - T_1}$$

Given the spot rate R for maturity T , the *instantaneous forward rate* for maturity of T is then

$$R_F = R + T \frac{\partial R}{\partial T}$$

If $P(0, T) = e^{-RT}$ is the price of zero-coupon bond maturity at time T , the equation is then

$$R_F = -\frac{\partial}{\partial T} \ln P(0, T)$$

Definition 6.1.7. *Forward Rate Agreement (FRA)* is an agreement to exchange a predetermined rate for a reference rate that will be observed in the market at a future time.

Let R_K be the fixed rate, R_F be the current forward rate for the reference rate, τ be the period of time to which the rates apply, L be the principal in the contract.

For the party that receives the fix rate, the FRA has a present value of

$$\tau(R_K - R_F)L$$

For the party that pays the fix rate, the FRA has a present value of

$$\tau(R_F - R_K)L$$

Definition 6.1.8. The *Duration* of a bond is a measure of how long the holder of the bond has to wait before receiving the present value of the cash payments.

Let c_i be cash flow at time t_i ($1 \leq i \leq n$). Bond price B , yield y (continuously compounded) are related by

$$B = \sum_{i=1}^n c_i e^{-y t_i}$$

The Duration D of the bond is then

$$D = \sum_{i=1}^n t_i \left[\frac{c_i e^{-y t_i}}{B} \right]$$

where the term in square brackets is ratio of present value of cash flow at t_i to bond price. Duration is hence a time-weighted average of the times when payments are made.

The relationship between duration and yield is as follows:

$$\begin{aligned} \Delta B &= \frac{dB}{dy} \Delta y \\ \frac{\Delta B}{B} &= -D \Delta y \end{aligned}$$

If y is expressed with compounding frequency of m times a year, then the relationship becomes

$$\Delta B = -\frac{BD \Delta y}{1 + y/m}$$

Hence, the *modified duration* is

$$D^* = \frac{D}{1 + y/m}$$

The duration relationship is then

$$\Delta B = -BD^* \Delta y$$

When duration is used for bond portfolios, it is assumed that the yields of all bonds will change by approximately the same amount, i.e., a parallel shift in the spot yield curve. The portfolio may still be exposed to shifts that are either large or non-parallel.

Convexity may be used to improve the relationship in the equation. Convexity is defined as

$$C = \frac{1}{B} \frac{d^2 B}{dy^2}$$

By Taylor series, we then have the relationship

$$\frac{\Delta B}{B} = -D\Delta y + \frac{1}{2}C(\Delta y)^2$$

To determine the underlying reasons for the shape of the spot curve, theories on term structure of interest rates are referred to.

(Expectations Theory) A forward interest rate corresponding to a certain future period is equal to the expected future zero interest rate for that period.

(Market Segmentation Theory) There exists no relationship between short, medium, and long-term interest rates, as investors do not readily switch from one maturity to another.

(Liquidity Preference Theory) Investors prefer to preserve their liquidity and invest funds for short periods of time. Borrowers prefer to borrow at fixed rates for long periods of time. Hence forward rates are greater than expected future spot rates.

6.2 Forwards and Futures

Based on the book by John Hull (2021).

6.2.1 Pricing

To examine how forward prices and futures prices are related to spot price, we assume the following are true for some market participants:

- i. No transaction costs
- ii. Same tax rate on all net trading profits
- iii. Money borrowed and lend are at the same risk-free rate
- iv. Arbitrage opportunities are taken advantaged of as they occur

Let T be time until delivery date (in years), S_0 be price of underlying asset today, F_0 be price of forward or futures today, r be zero-coupon risk-free rate per annum in continuous compounding (maturing in T years).

Consider a forward contract on underlying asset with spot price S_0 that provides no income. Then

$$F_0 = S_0 e^{rT}$$

If $F_0 > S_0 e^{rT}$, long asset and short forward. If $F_0 < S_0 e^{rT}$, short asset and long forward.

If short sales are not possible, and arbitrage opportunities exist, then if $F_0 > S_0 e^{rT}$, investor may:

1. Borrow S_0 in cash at an interest rate r for T years
2. Buy 1 unit of asset
3. Enter forward contract to sell 1 unit of asset

At time T , asset is sold for F_0 , investor to repay $S_0 e^{rT}$ loan, making profit $F_0 - S_0 e^{rT}$.

If $F_0 < S_0 e^{rT}$, investor may:

1. Sell asset for S_0
2. Invest proceeds at interest rate r for time T
3. Enter into forward contract to buy 1 unit of asset

At time T , cash has grown to $S_0 e^{rT}$. Investor repurchase asset for F_0 , makes profit of $S_0 e^{rT} - F_0$.

If the underlying asset provide income with present value of I during life of forward, then

$$F_0 = (S_0 - I)e^{rT}$$

If $F_0 > (S_0 - I)e^{rT}$, investor may long asset and short forward. If $F_0 < (S_0 - I)e^{rT}$, investor may short asset and long forward. If short sales are not possible, investors owning the asset will sell the asset and long forward.

If the underlying asset provides a known yield rather than income, with q as the average yield, then the following strategy must generate zero profit to prevent arbitrage:

1. Borrow S_0 to buy one unit of asset at time 0
2. Enter into forward to sell e^{qT} units of asset at time T for F_0
3. Close the forward by selling e^{qT} units of the asset at time T

Hence we have

$$S_0 e^{rT} = e^{qT} F_0$$

or

$$F_0 = S_0 e^{(r-q)T}$$

When a forward contract is first entered, the value is close to zero. Let K be delivery price negotiated some time ago, with T years delivery date, r is T -year risk-free interest rate, F_0 is forward price if contract is negotiated today. Let f be value of forward contract today. Then

$$f = (F_0 - K)e^{-rT}$$

In the case of stock indices, if $F_0 > S_0 e^{(r-q)T}$, profit can be made by buying stocks underlying the index at spot price and shorting the index futures contract. If $F_0 < S_0 e^{(r-q)T}$, short the stocks and long futures. This is known as *index arbitrage*.

In the case of currencies, let r_f be foreign risk-free rate in a foreign-denominated bond, S_0 be spot price of local currency in foreign currency, F_0 be forward or future price of local currency in foreign currency. Then the *interest rate parity* relation persists in the form

$$F_0 = S_0 e^{(r-r_f)T}$$

For commodities, let U be present value of storage costs net of income during life of forward. Then

$$F_0 = (S_0 + U)e^{rT}$$

If storage costs are proportional to price of commodity, they can be treated as negative yield. Let u be storage costs per annum as proportion of spot price net of any yield earned on asset. Then

$$F_0 = S_0 e^{(r+u)T}$$

If $F_0 > (S_0 + U)e^{rT}$, then investor may take advantage of arbitrage:

1. Borrow $S_0 + U$ at risk-free rate, purchase one unit of commodity and pay storage costs
2. Short futures on one unit of commodity

If $F_0 < (S_0 + U)e^{rT}$, then investor may take advantage of arbitrage:

1. Sell commodity, save storage costs, invest proceeds at risk-free interest rate
2. Long futures contract

Benefits from holding physical assets are *convenience yields*, such as by crude oil manufacturers. Let y be the convenience yield, then

$$F_0 e^{yT} = (S_0 + U)e^{rT}$$

If storage costs per unit are a constant proportion of spot price, then y is defined such that

$$F_0 e^{yT} = S_0 e^{(r+u)T}$$

Convenience yield reflects market expectation on future availability of the commodity. The greater the possibility that shortages will occur, the higher the commodity yield.

The *cost of carry* measures storage cost plus interest paid to finance the asset less income earned:

- i. Non-dividend paying stock: r , as no storage and income is earned
- ii. Stock index: $r - q$, as income is earned at rate q on asset
- iii. Currency: $r - r_f$
- iv. Commodity: $r - q + u$, where it provides income at rate q and requires storage costs at rate u

6.2.2 Hedging with Futures

The fundamentals of hedging with futures are *hedge-and-forget* strategies, where no changes is made to adjust the hedge once it has been put in place.

Definition 6.2.1. (Basic Principles of Futures Hedging)

The objective is to take a position that neutralises the risk as far as possible.

- i. *Short Hedge*: short position on futures.
Used when hedger already owns an asset and will sell the asset at some time in the future; or when asset is not owned right now but will be owned and ready for sale sometime in the future.
- ii. *Long Hedge*: long position on futures.
Used when hedger will purchase an asset in the future and wants to lock in the price now.

In practice, hedging is not perfect due to factors as follows:

1. Asset being hedged is not exactly the same as the asset underlying the futures contract.

	Short Hedge	Long Hedge
May 15	Spot: 50 Futures: 49	Spot: 50 Futures: 49
August 15 Scenario 1	Spot: 45 Gain from hedge: 4	Spot: 45 Loss from hedge: 4
August 15 Scenario 2	Spot: 55 Loss from hedge: 6	Spot: 55 Gain from hedge: 6

2. Uncertainty as to exact date in which the asset will be bought or sold.
3. Hedge may require the futures contract to be closed out before its delivery month.

These lead to *basis risk*.

Definition 6.2.2. The *basis* in a hedging situation is defined as

$$\text{Basis} = \text{Spot Price} - \text{Futures Price}$$

An increase/decrease in basis is a strengthening/weakening of the basis.

Definition 6.2.3. Let S_i be spot price at time t_i , F_i be futures price at time t_i , b_i be basis price at time t_i . Assume hedge is placed at time t_1 , closed at time t_2 . Price realised for asset is S_2 , profit from futures position is $F_1 - F_2$. Effective price obtained for asset hedging is therefore $S_2 + F_1 - F_2 = F_1 + b_2$. If b_2 is known, perfect hedge will result. The *basis risk* is the hedging risk from uncertainty associated with b_2 .

Definition 6.2.4. *Cross Hedging* occurs when the asset underlying the futures contract is not the same as the asset whose price is being hedged.

Cross hedging is often used when futures of the original asset being hedged are not actively traded on the market, and the hedger seeks an alternative asset to hedge the original asset.

Definition 6.2.5. *Hedge Ratio* is the ratio of size of position taken in futures contract to the size of exposure.

Assuming no daily settlement of futures contracts, hedger seeks a hedge ratio that minimises variance of hedged position value. Let ΔS be change in spot price, ΔF change in futures price. Assuming linear relationship,

$$\Delta S = a + b\Delta F + \epsilon$$

where a, b are constants, ϵ is an error term. Suppose hedge ratio is h . Change in value of position per unit of exposure to S is:

$$\Delta S - h\Delta F = a + (b - h)\Delta F + \epsilon$$

Standard deviation is minimised by setting $h = b$. Let minimum variance hedge ratio be h^* . Then

$$h^* = \rho \frac{\sigma_S}{\sigma_F}$$

where σ_S, σ_F is standard deviation of $\Delta S, \Delta F$ respectively, ρ is coefficient of correlation.

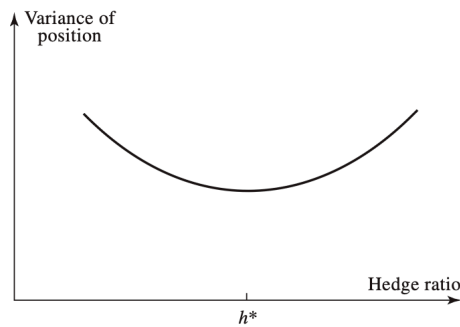


Figure 8: Dependence of variance of position on hedge ratio.

Hedge effectiveness is the proportion of variance eliminated by hedging. This is R^2 from regression of ΔS against ΔF , and equals ρ^2 . Parameters ρ, σ_S, σ_F are estimated from historical data on ΔS and ΔF .

The optimal number of futures to be used in hedging is

$$N^* = \frac{h^* Q_A}{Q_F}$$

where Q_A is size of portion being hedged (units), Q_F is size of one futures contract (units). The futures contract should be on $h^* Q_A$ units of the asset.

If daily settlement is used, there are a series of one-day hedges, and thus let $\hat{\sigma}_S, \hat{\sigma}_F$ be standard deviation of percentage one-day changes in spot and future price respectively, $\hat{\rho}$ be correlation between percentage one-day changes in spot and future prices. The optimal one day hedge is then

$$h^* = \hat{\rho} \frac{\hat{\sigma}_S S}{\hat{\sigma}_F F}$$

and the optimal number of futures to be used is then

$$N^* = \hat{\rho} \frac{\hat{\sigma}_S S Q_A}{\hat{\sigma}_F F Q_F}$$

If an interest $r\%$ per annum is earned or paid over the remaining life of the hedge, then the optimal number of futures is $N^*/(1 + 0.01r)$; this is *tailing the hedge*.

Stock index futures may be used to hedge a well diversified equity portfolio. Let V_A, V_F be the current value of portfolio and one futures contract respectively.

If portfolio mirrors the index, the optimal hedge ratio is then 1.0, and number of futures contracts to be shorted is then $N^* = \frac{V_A}{V_F}$. If portfolio do not mirror the index, then capital asset pricing model (CAPM) should be used to determine beta (β), and the number of futures contracts to be shorted is then $N^* = \beta \frac{V_A}{V_F}$, assuming maturity of futures contract is close to the maturity of the hedge.

If instead, the hedger wishes to change the beta of portfolio where $\beta > \beta^*$, a short position $(\beta - \beta^*) \frac{V_A}{V_F}$ is required. If $\beta < \beta^*$, then a long position $(\beta^* - \beta) \frac{V_A}{V_F}$ is required.

Stock index hedging is typically used when the portfolio manager is uncertain about performance of market, but is confident that the stocks in the portfolio will outperform the market. The hedger may also be planning to hold a portfolio for a long period of time and requires short-term protection in an uncertain market situation.

If expiration date of hedge is later than delivery dates of all futures contracts that may be used, then the hedger may *stack and roll* by closing out one futures contract and taking the same position in a futures contract with a later delivery date.

6.2.3 Interest Rate Futures

The *day count* defines the way in which interest accrues, and is expressed as X/Y , where X is the way in which number of days between two dates is calculated, Y is total number of days in reference period.

$$\text{interest earned between two dates} = \frac{X}{Y} \times \text{interest earned in reference period}$$

The three common day count conventions used in United States are:

- i. Actual/Actual (in period): for Treasury Bond
- ii. 30/360: for corporate and municipal bonds, assumes 30 days a month and 360 days a year
- iii. Actual/360: for money market instruments

Prices of money market instruments are quoted using discount rate, which is interest earned as a percentage of final face value. Let P be quoted price, Y be cash price, n remaining life of Treasury bill in calendar days:

$$P = \frac{360}{n}(100 - Y)$$

Prices of treasury bonds are quoted in dollars and thirty-seconds of a dollar, i.e., 120-05 or $120\frac{5}{32}$, which is a bond value of \$120, 156.25. The quoted price is *clean price*, and the cash paid by purchaser is *dirty price*:

$$\text{cash price} = \text{quoted price} + \text{accrued interest since last coupon date}$$

7 Currency Trading

8 Commodities Trading

9 Appendix

For everything that cannot be classified under algorithmic trading, but is absolutely necessary for executing the concepts.

9.1 Visual Studio Code

Based on the book by April [Speight \(2021\)](#).

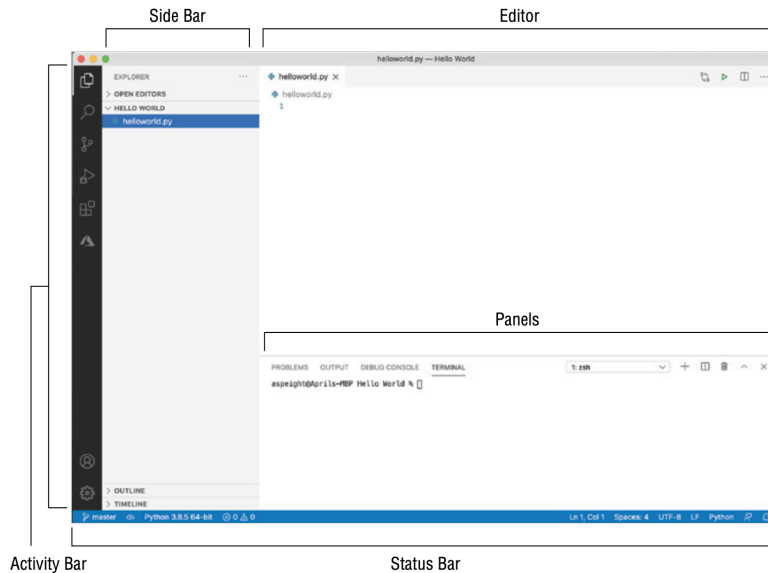


Figure 9: The visual studio code interface.

Activity bar on the far-left side allows switching between views, with quick access to tasks such as:

- i. Explorer: file and folder management
- ii. Search: global search and replace, can use plain text or regex
- iii. Source Control: Git source control
- iv. Run: for debugging, such as variables, call stacks, breakpoints
- v. Extensions: browsing, installation, management of extensions

Activity bar may also provide custom views from extensions installed from Extension Marketplace. Views may be hidden, dragged and dropped around.

Side bar displays active view. Editor where files may be edited, may be resized, and top editor region changes depending on the type of file active in the editor; the top bar may also have a Source Control view if project is connected to git; files may be pinned and grouped by tabs.



Figure 10: Source control button

Panels are for debugging information, errors and warnings; and also for opening an integrated terminal on the root of the project. A REPL terminal (Python standard shell) may also be opened.

The status bar contains information about the opened project and files being edited. This includes: source control management with Git, total number of problems for the opened programs, line/column, indentation settings for space or tabs, encoding setting, end-of-line sequence setting, language mode, VS Code feedback mechanism, and notifications.

The command palette at the very top of the UI can be used to run commands to execute editor tasks in addition to extension commands.

9.1.1 Setting Up Python

To set default interpreter path in VS Code, in settings editor, search for `python.pythonPath`. In Python: Default Interpreter Path setting, enter the path to the interpreter.

To enable Quick Fix which help fix issues identified by warnings or errors (with lightbulb popping up), in settings editor, search for `python.jediEnabled`, then set it to false.

IntelliSense is a variety of tools to assist with programming, such as code completion, object definition, location of object or variable declarations. These are triggered by either pressing Ctrl+Spacebar, or by typing a trigger character (i.e., a dot character in Python).

If linter detects any errors, these will be present in the Panels' Problem tab.

Refactoring is used to maintain functionality while improving the internal structure or architecture of a program. This should be a routine task that occurs before any updates or new features are added to a program. VS Code can help with refactoring via the following commands in Command Palette: Extract Variable, Extract Method, Sort Imports. Refactoring requires the `Rope` library.

Extract variable command allows extracting all similar occurrences of the same constant value of expression in multiple places, and replaces it with a variable. May be accessed via Python Refactor: Extract Variable.

Extract method command extracts all similar occurrences of selected expression or block, creates a new method, and replaces the expression with a method call. May be accessed via Python Refactor: Extract Method.

Sort imports method uses `isort` packages to consolidate specific imports from the same module into a single import statement, and organises 'import' statements in alphabetical order.

If a code pattern is repeated within a file or across multiple files, code snippets may be used, by searching in Command Palette for Snippets: Configure User Snippets.

9.1.2 Debugging

After starting debug session, the Run view opens. As the debugger runs, the current state of variables is reflected in Variables panel, which organises variables into local and global scopes. In the editor, the Debug toolbar will appear with the following functionalities:

- i. Continue (A): Runs all the code after the breakpoint up to the next breakpoint or end of program.
- ii. Step Over (B): Step line by line at the current scope. If current line is a function call, debugger runs the function entirely and then pauses at the next line after function call.
- iii. Step Into (C): Steps over each line within the function scope and any additional function calls.
- iv. Step Out (D): To exit from within a function to the scope that called it.
- v. Restart (E); Stops execution and restarts the debugging session.
- vi. Stop (F): Stop all execution without finishing the program.

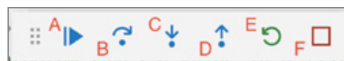


Figure 11: Debug toolbar on Editor

The side bar also has a call stack section, which shows the whole chain of function calls leading up to the current point of execution. Useful if calls go through other files in the project.

Conditional breakpoint allows for stopping of the code during specific conditions to evaluate and debug.

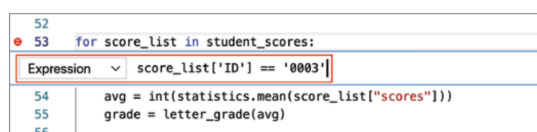
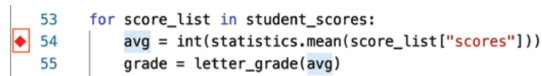


Figure 12: Adding a condition in a breakpoint

A logpoint outputs message to Debug Console without breaking the debugger. Expressions can be evaluated within curly braces.



```
53 for score_list in student_scores:
54     avg = int(statistics.mean(score_list["scores"]))
55     grade = letter_grade(avg)
```

Figure 13: Adding a logpoint

If program has a lot of variables which is hard to keep track of in variables pane, the variables may be added to the Watch panel which keep tracks of the variables.

The Debug Console allows access and modification of all the program's variables, call functions, evaluate expressions, and whatever code in the program's current state, rather than modifying the code and restarting. Different scenarios may be tested in the Debug Console and copy the fix into the program while the debugger is paused. The Debug Console also shows suggestions as code is typed.

To enable a test framework, *pytest* or *unittest* must be installed. Run the command Python: Configure Test to select the framework, directory that contains the test, and pattern to identify test files.

- i. unittest: Looks for any Python file with 'test' in the name in top-level project folder. All test files must be importable modules or packages. To specify discovery pattern, change the pattern in *python.testing.unitTestArgs*.
- ii. pytest: Looks for any Python file beginning with 'test_' or ends with '_test' located anywhere within current folder and all subfolders.

References

- Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.
- Hull, J. C. (2021). *Options, Futures, and Other Derivatives*. Pearson.
- Narang, R. K. (2013). *Inside the Black Box: The Simple Truth About Quantitative Trading*. Wiley Finance Series. Wiley.
- Prado, M. L. D. (2018). *Advances in Financial Machine Learning*. Wiley Finance Series. Wiley.
- Speight, A. (2021). *Visual Studio Code for Python*. Wiley.
- Thierry, A. and G. Helyette (2000, October). Order flow, transaction clock, and normality of returns. *The Journal of Finance* 55(5), 2259–2284.
- Velu, R. (2020). *Algorithmic Trading and Quantitative Strategies*. CRC Press.