Advanced Python for Finance

Lecture 5

Lecture 5 Agenda

1. Tolerance as Function - recap
2. Implementation Shortfall with Tolerance Function
3. Tolerance as a Function
4. Implementation Shortfall with a Tolerance Factor
5. Lee-Ready Trade Direction Indicator
6. A Simple Pairs Algo

Tolerance Factors

Tolerance Factors

We have now established the concept of target functions.

The next step is to consider how we will allow ourselves to vary from the deterministic target function over time in order to take advantage of market fluctuations, short term forecasts, etc.

We will define the concept of **tolerance factor.**

Tolerance Factors

Establishing a Tolerance factor:

- How far can you get behind (or ahead)
- Think of this factor as a proxy for **risk tolerance**

Simple intuition:

- Tolerance is high = patient. Capture spread, optimize for cost
- Tolerance is low = aggressive. Stay close to target, pay spread. Risk averse

Threshold model A:

1. Calculate a threshold value, when you are behind by some amount > threshold value then trade aggressively until amount behind < threshold

2. Threshold value = % behind
   % behind = (1- actual) / target

Implementation Shortfall

A Simple Implementation Shortfall Strategy

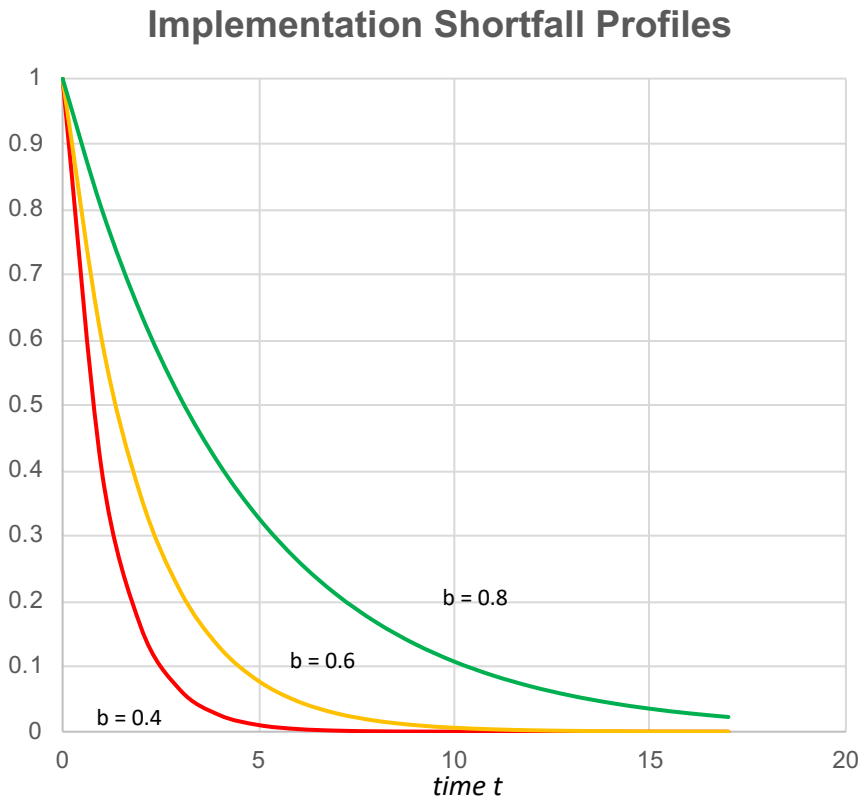
$$y = 1 - b^t$$

where

y = % of order complete at time *t*

b = risk tolerance 0 <= b <= 1,

t = minutes from start



Opportunity cost / impact trade off

- Common parameters
  - Volume caps
  - Aggressiveness / patience (tuning the tradeoff)
- Models vs. heuristics
- Cleanup logic

Implementation Shortfall

5.1 Implementation Shortfall.ipynb

Theoretical Values and

Multi-factor Price Models

Rules and Multiple Signals

Assume we are buying and consider if we have a model with three signals

Signal 1

```
if current_shares_behind <= allowed_behind: # we're passive
        buy at bid

else: # we're behind, get aggressive
        buy at offer
```

Signal 2

```
if momentum_signal > momentum_threshold: # trade aggressively
        buy at offer or sell at bid
```

Signal 3

```
if reversion_signal >= reversion_threshold: # trade passively
        buy at bid
```

Combinations of Factors

Before we get to linear combinations, let's consider multiple factors in a simple "voting" mechanism.

Example Assumptions:

1. Consider a VWAP **buy** order
2. We can only buy on the bid (passive) or the offer (aggressive). Signal must be +0.5 to be aggressive
3. Assume equal weighting of all factors

Case	Schedule	Momentum	Reversion	FV adjustment
1	On schedule: 0	Tick up: +0.5	Neutral: 0	+0.5 aggressive
2	Very behind: +0.5	Tick down: -0.5	Neutral: 0	0.0: passive
3	Very behind: +0.5	Tick down: -0.5	Down: -0.5	-0.5 passive
4	A little behind: +0.2	Tick up (weaker) +0.2	Up (weak): +0.1	+0.5 aggressive

This gets even more interesting for non-execution strategies with bi-directional signals!

Multiple Signals and Theoretical Values

We need to be able to combine multiple factors into one expression of a desire to trade at a given price (passive, aggressive, or anywhere in between)

Applicable to any signal, not just tick-level / microstructure

Tolerance Factors: Model B

- Apply adjustment as a continuous factor
  - Combines with other factors
  - Can support concept of “ahead” as well as “behind” (i.e. if my short term signals are strong, can I trade ahead to some degree?)
- Use % ahead/behind to adjust target price continuously
  - Start with a mid or “micro” price P_m = midpoint of spread
  - Apply “catchup” factor F_c such that $0.0 < F_c < (Q_{ask} - Q_{bid})$:
  - If I’m buying, and not behind, my catchup factor is 0. buy at bid
  - If I’m buying and sufficiently behind, let’s say my catchup factor is 1.0. Buy at offer
  - If I’m only slightly behind, F_c might be 0.2
  - Buy Order Placement: “Fair” price $P_f = P_m – ((Q_{ask} - Q_{bid}) / 2) + F_c$
  - $F_c = k(Q_{ask} - Q_{bid})$
  - k = c * (target - actual) / total shares
  - c = calibration factor – tune based on empirical performance

Arbitrage Pricing Theory

Consider Arbitrage Pricing Theory (APT)

(Ross, 1976), A decent intro at: https://en.m.wikipedia.org/wiki/Arbitrage_pricing_theory

A stock's return can be expressed as a *linear combination* of multiple factors:

$$r_j = a_j + b_{j1}F_1 + b_{j2}F_2 + \cdots + b_{jn}F_n + \epsilon_j$$

Where:

a_j = constant factor for stock j

F_k = factor

b_{jk}= stock *j's* sensitivity to factor F_k

Fair Value

How does this help us?

We can add arbitrarily many factors in linear combination to arrive at a predicted return and a "fair value" or "*FV*" of a stock.

$$FV = s_j + r_j$$

Factors can be

- *signals* such as tick test, order imbalance
- *risk-based*
- or anything else you can think of

Fair Value Factors - Intuition

When we think something is over priced we sell:

Lower FV versus the market

When we think something is under priced, we buy:

Raise FV so stock looks undervalued by the market

Risk-Based Factor Example: Schedule

Consider a schedule based strategy.

"Risk" is defined as the difference between your current position and your target position. The greater the difference (i.e. the greater behind), the greater your risk

If you are buying:

1. If you are behind, you want to buy more: raise FV

2. If you are ahead, you want to buy less: lower FV

3. If you are on schedule, you are neutral. FV at midpoint

Risk-Based Factor Example: Schedule

And again we can again express this the signal as a continuous function:

$$FV = midpoint_j + b_{jF}F_j$$

Where

b_{jF} = loading of the tick signal factor

F_j = schedule factor described in spread units in some range, e.g. [-0.5..0.5]

- State 1: behind (buying): FV = the offer price = midpoint + 0.5 # trade now
- State 2: ahead (buying): FV = the bid price = midpoint – 0.5 # slow down
- State 3: on schedule: FV = midpoint + 0 # no impact
- States between 1 and 2: FV = midpoint + F_j spread units

- Reverse signs for selling

Risk-based Factor Example: Position

Consider a proprietary trading strategy in which a variety of signals are working.

In the absence of information you would prefer to *not* have a position:

1. If you are long you would prefer to be flat (sell to close)

2. If you are short, you would prefer to be flat (buy to close)

Risk-Based Factor Example: Position

We can again express this the signal as a continuous function:

$$FV = midpoint_j - b_{jF}F_j$$

Where

b_{jF} = loading of the position factor

F_j = risk adjustment factor described in spread units in the range [–0.5..0.5]

- State 1: Long (sell): FV = the bid price = midpoint – 0.5
- State 2: Short (buy): FV = the offer price = midpoint + 0.5
- State 3: 0 position: FV = midpoint + 0
- States between 1 and 2: FV = midpoint + F_j spread units

Putting it together for VWAP

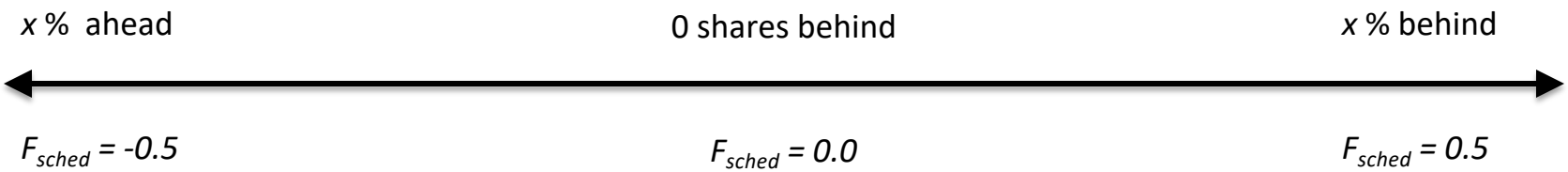
$$FV_j = midpoint_j + b_j F_{tick} + b_j F_{imbal} + b_j F_{schedule}$$

Where for stock *j*:

F_k = factor for each signal or adjustment

b_{jk}= stock *j's* sensitivity to factor F_k

VWAP v2 Schedule Factor Construction



$$F_{schedule} = \frac{q_{pct\_behind}}{q_{\max\_behind}} \times \frac{spread_{avg}}{2}$$

Example 1: If max_behind = 0.02 and average spread = $0.05, then:

At 0.02 behind: $\frac{0.02}{0.02} \times \frac{0.05}{2} = 0.025$

Example 2: If max_behind = 0.02 and average spread = $0.05, then:

At 0.01 behind: $\frac{0.01}{0.02} \times \frac{0.05}{2} = 0.0125$

VWAP v2 Fair Value Calculation

$$FV_j = midpoint_j + b_j F_{schedule}$$

Where:

F_k = factors for each signal or adjustment

b_{jk}= stock *j's* sensitivity to factor F_k

Example 1 continued:

if bid = 100, offer = 100.05, max_behind = 0.02 and avg_spread = 0.05

midpoint = 100.025

$F_{schedule}$ = 0.025

FV = 100.025 + 0.025 = **100.05** **:** this is >= offer price, so aggress

What happens if current spread is < or > average spread??

VWAP v2 Order Placement

$$FV_j = midpoint_j + b_j F_{schedule}$$

It's time to place our order. Even though our FV should now reflect our "behindness", we still may not necessarily want to place the order at that price in call cases.

Why?

Simple placement logic for order_side = BUY:

```
if FV >= offer:
    order_price = offer
else # FV is not sufficiently high to warrant an aggressive order
    order_price = bid
```

Signal Factor Example: Tick Test

How do we use the theoretical value in trading decisions?

1. Consider a simplified conceptual version of a tick test signal T_j
2. Derive a signal from autocorrelation of trade directions:
   - State 1: Next tick predicted up
   - State 2: Next tick predicted down
   - State 3: Next tick indeterminate
3. Assume for the moment we want to trade immediately based on that signal
4. For a stock j, define price sj to be the midpoint within the current quote.

   We can then express our three states as:
   - State 1: Next tick predicted up (buy): FV = the offer price = midpoint + spread/2
   - State 2: Next tick predicted down (sell): FV = the bid price = midpoint – spread/2
   - State 3: Next tick indeterminate: FV = midpoint

Signal Factor Example: Tick Test

More generally, we can express this the signal as a continuous function:

$$FV = midpoint_j + b_{jF}F_j$$

Where

b_{jF} = loading of the tick signal factor

F_j = tick signal factor described in spread units in the range [–0.5..0.5]

- State 1: Next tick predicted up (buy): FV = the offer price = midpoint + 0.5
- State 2: Next tick predicted down (sell): FV = the bid price = midpoint – 0.5
- State 3: Next tick indeterminate: FV = midpoint + 0
- States between 1 and 2: FV = midpoint + F_j spread units

IS Algos with Factors

5.2 IS v2.ipynb

Lee-Ready: Trade Identification

Trade Identification

Inferring Trade Direction from Intraday Data

Charles M. C. Lee; Mark J. Ready

The Journal of Finance, Vol. 46, No. 2 (Jun., 1991), 733-746.

- All Trades are a match between one or more buy orders and one or more sell orders.
- Refer back to our concepts of passive (providing) and aggressive (taking)
- The liquidity taking order will cause the the trade to occur
- The question: can we identify "directionality" of trades, i.e. did a buy or a sell order trigger the trade?
- If we can identify buys and sells from trade data then we can potentially use that information in our analyses and trading signals

Trade Identification

We will consider three general cases

1. Identifying Trades without Quotes
2. Identifying Trades with Quotes
3. Trades inside the Spread

Trade Identification without Quotes

- We often may have only trades, not quotes
- In these cases, we can derive information from the trades themselves
- Let's classify trades relative to the trades immediately preceding them by two methods:
  - Tick Test
  - Reverse Tick Test

Tick Test

Look at trade price compared with the immediately preceding trade:

1. The trade is an **uptick** if the price is *higher* than the previous trade.
2. The trade is a **downtick** if the price is *lower* than the previous trade.
3. The trade is a **zero-uptick** if the previous trade was at the same price, but the previous price change was an *uptick*.
4. The trade is a **zero-downtick** if the previous trade was the same price, but the previous price change was a *downtick*.

Classification

A trade is a **buy** if it occurred on an uptick or a zero-uptick

A trade is a **sell** if it occurred on an downtick or zero-downtick

Reverse Tick Test

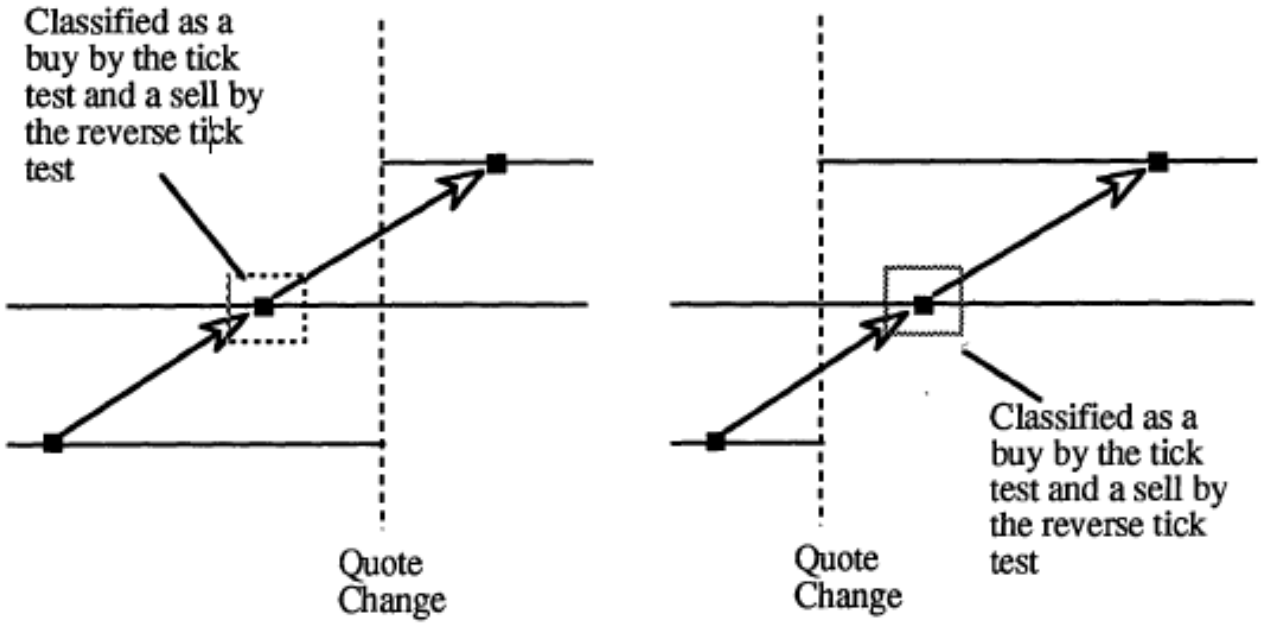
The reverse tick test compared to the immediately **following** trade.

1. If the following trade is at a *higher* price, the current trade is classified as a **sell**.

2. If the following trade is at a lower price, the current trade is classified as a **buy.**

Tick Test and Reverse Tick Test

Classification of Trades in Trending Markets with Quote Changes



Source: Lee, Ready (1991)

Tick Test and Reverse Tick Test

Table I

A Comparison of the Tick Test and the Reverse Tick Test When Identification of the Prevailing Quote is Unambiguous

For a sample of 150 NYSE firms during 1988, this table reports the percentage of trades classified as buys and sells by the tick test and the reverse tick test when the identity of the prevailing quote is unambiguous. A trade is classified as a buy (sell) by the tick test if the *prior* price change is positive (negative). A trade is classified as a buy by the reverse tick test if the *next* price change is negative (positive). A prevailing quote is considered unambiguously identified if the most recent quote revision occurred more than 5 seconds before the trade. This situation applied to 90.9 percent of the classified (nonopening) trades in the sample.

Classification Based on the Prevailing Quote	Tick Test Classification	Reverse Tick Test Classification
Buy—Above Ask	98.8% buy	91.3% buy
Buy—At Ask	92.1% buy	80.0% buy
Inside the Spread	52.4% sell	52.0% buy
Sell—At Bid	90.2% sell	79.4% sell
Sell—Below Bid	98.7% sell	94.5% sell

Source: Lee, Ready (1991)

Trade Identification with Quotes

We may have quotes, but which ones are the right ones to use?

Lee-Ready asks: are quotes at the time of a trade the *cause* or the *result* of a trade?

Using their *isolated trades*, they established that if a quote is *less than 5 seconds old*, it was probably caused by the trade being examined.

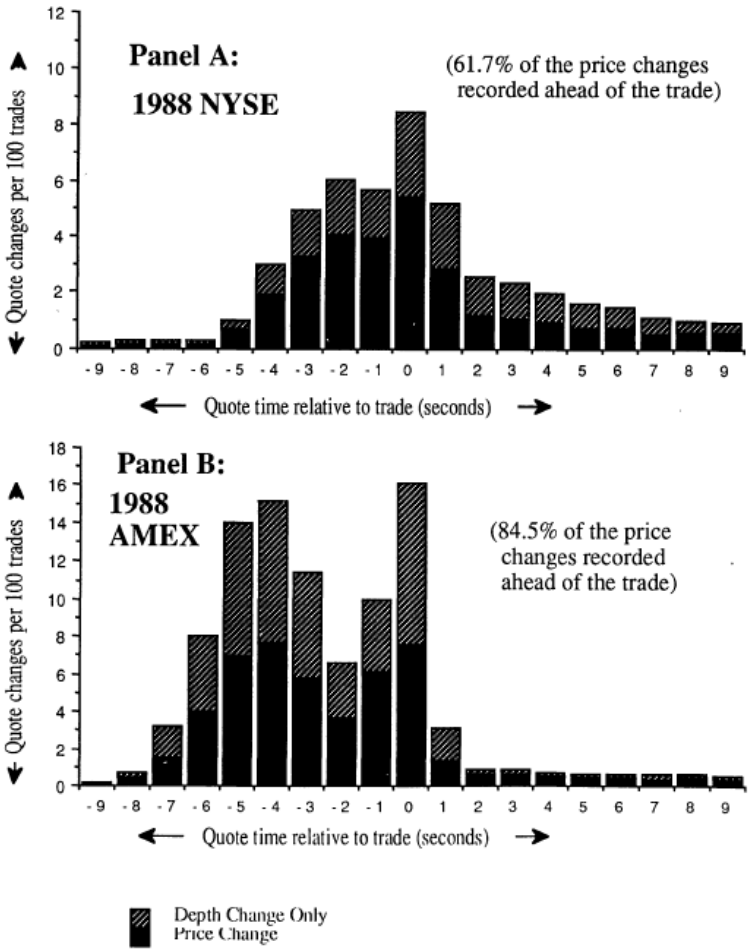


Figure 2. The frequency of quote revisions around isolated trades. The frequency of quote revisions in a 19-second window centered on isolated trades for 150 NYSE firms and all AMEX firms during 1988. An isolated trade is defined as the first trade after 11:00 a.m. but before 2:30 p.m., with no other trades within a 2-minute window centered on the trade. Both quote revisions which reflect price changes and quote revisions which affect depth changes only are shown.

Source: Lee, Ready (1991)

Trades Inside the Spread

Proposed options

- Location relative to quote ("bidness" and "askness")
  - A trade is a **buy** if the price is closer to the ask
  - A trade is a **sell** if the price is closer to the bid
  - Trades at midpoint are discarded

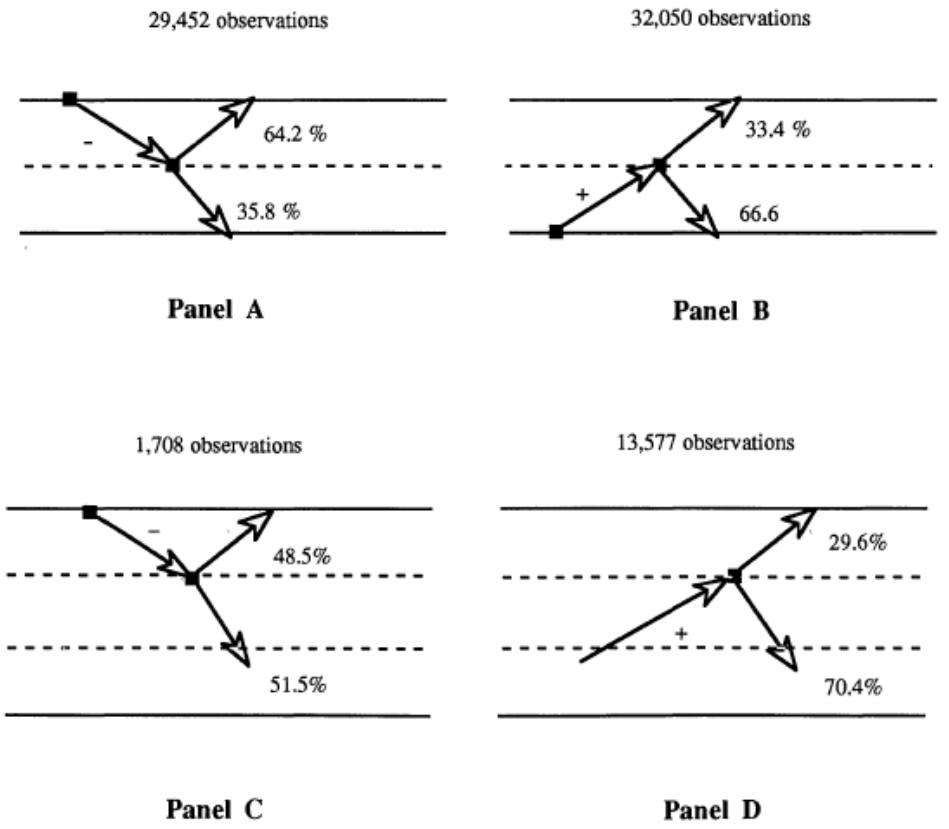


Figure 3. Frequency of price reversals for trades inside the spread. For a sample of 150 NYSE firms during 1988, these figures report the frequency of trades at the midpoint of a 1/4 spread (Panels A and B) and at the upper inside point of a 3/8 spread (Panels C and D). Panels A and C show the number of trades which occurred on a downtick and the proportion of uptick and downtick movements in the next price change. Panels B and D show the number of trades which occurred on an uptick and the proportion of uptick and downtick movements in the next price change.

Source: Lee, Ready (1991) *(NOTE: ¼ spread = $0.25, 3/8 spread = $0.375)*

Trades at Midpoint

According to the scenarios put forth in the paper, the tick test is effective at classifying trades inside the spread as well.

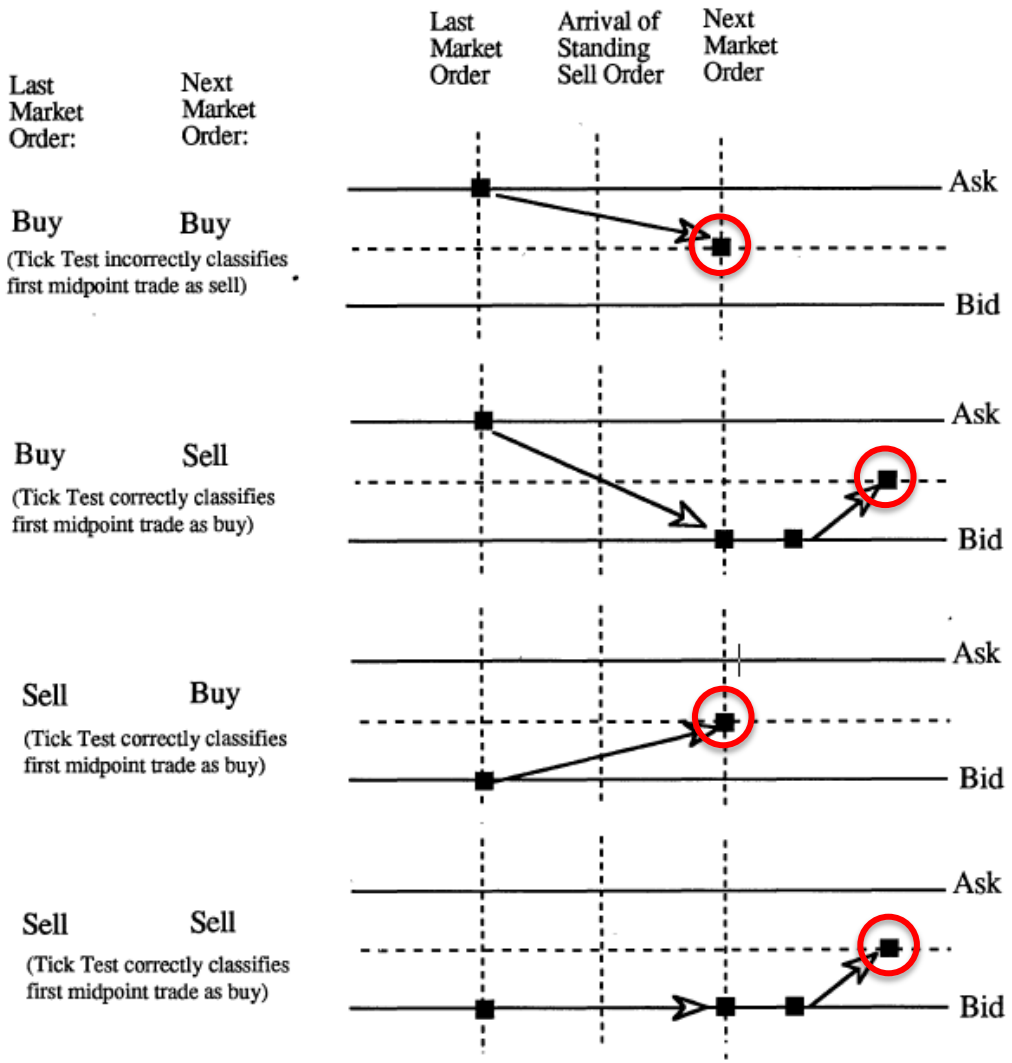


Figure 4. Classifying the first midpoint trade after the arrival of a standing sell order. Assume constant bid and ask prices and a ¼ spread. Further assume market buy and sell orders follow independent Poisson processes with the same arrival rate. Standing buy and sell orders also follow independent Poisson processes with a lower average arrival rate than market orders. The four equally likely patterns of market orders immediately before and after the arrival of a standing order are depicted above (buy/buy, buy/sell, sell/buy, and sell/sell). In three of the four cases, the tick test classification for the first midpoint trade is correct.

Source: Lee, Ready (1991)

Summary of the Lee-Ready Algorithm

1. If quotes are available, use them:
    a) If quotes are < 5 seconds (!) old, use previous quote
    b) Trades at the bid are **sells**
    c) Trades at the offer are **buys**
2. If trade is inside the spread but not at midpoint, classify by proximity to the bid or offer, apply same logic as #1
3. If trade is at midpoint or no quotes are available, use the tick test.

See the original paper for more detail

Limitations of Lee Ready

Changing Market Structure

- Narrower spreads
- Decimalization (2001)
- The original paper's 5-second window!
- Replication with isolated trades
- Market fragmentation
  - Non-displayed liquidity
  - Latency considerations

Signal Factor Example: Tick Test

How do we use the theoretical value in trading decisions?

1. Consider a simplified conceptual version of a tick test signal T_j
2. Derive a signal from autocorrelation of trade directions:
   - State 1: Next tick predicted up
   - State 2: Next tick predicted down
   - State 3: Next tick indeterminate
3. Assume for the moment we want to trade immediately based on that signal
4. For a stock *j*, define price s_j to be the midpoint within the current quote.

   We can then express our three states as:
   - State 1: Next tick predicted up (buy): FV = the offer price = midpoint + spread/2
   - State 2: Next tick predicted down (sell): FV = the bid price = midpoint – spread/2
   - State 3: Next tick indeterminate: FV = midpoint

Measuring Autocorrelation

Let's look at the autocorrelation of the tick data:

1. Load Tick Data, (just trades)

2. Apply Lee-Ready (tick test) to do up/down
   - Use shift() operator to obtain previous row
   - Calculate difference
   - Use numpy sign() operator to retrieve the sign for each record
   - Apply from tick 2 to n

3. Calculate autocorrelation for n lags
   - Use autocorr(lag=n) for specific lags

4. Examine and Plot

See "L4.6 Tick Test.ipynb"

Signal Factor Implementation: Tick Test

Once again, we can express this the signal as a continuous function:

$$FV = midpoint_j + b_{jF} F_{tick}$$

Where

b_{jF} = loading of the tick signal factor

F_{tick} = tick signal factor described as buy: 1, sell: -1

Calculate the exponential moving average of the tick value, something like this:

$$F_n = \frac{1}{\tilde{\tau}_w} f_n + w_n F_{n-1}$$

Where the weighting factor is defined as a function of a volume window $\tilde{\tau}_w$:

$$w_n = e^{-(t_n - t_{n-1})/\tilde{\tau}_w}$$

In practice you would calibrate the decay factor once and store the value.

f_n is by construction in {-1,1}, so multiply by $\frac{spread_{avg}}{2}$ to scale the factor.

Source: Derived from Michael Sotiropoulos BAML Lecture on trading models

Determining Volume Window

For the volume window, we want trade time, not clock time.

During simulation we can count trades tick by tick and construct a rolling window of *n* trades.

But if you want to convert to clock time:

$$\tilde{\tau}_w = max\left(\tau_{min}, \left(\tau_{max}, \frac{T_{day}}{N_{trd}} n\right)\right)$$

Where:

τ_{min}, τ_{max} are user defined min and max values

T_{day} = length of the trading day (390 minutes for the US stock exchanges)

N_{trd} = avg number of trades per day

n = target number of trades to include

Source: Derived from Michael Sotiropoulos presentation on trading models

Pairs

Pairs

5.3 Pairs.ipynb