Chapter X - Basic High Frequency Trading Examples

High frequency trading (HFT) strategies typically trade intraday from milliseconds to minutes. They process large volume of data in a very short-term time horizon, from which they generate algorithm-based trading signals and execute those signals in a high-speed fashion. The whole process from signal generation to implementation is computerized and besides the validitiy and strength of the specific signal a HFT strategy is utilizing, the speed to execute the signal vitally determines the strategy's outcome P/L. The popularity of HFT strategies depends on two pivotal conditions: 1) the availability of transaction-by-transaction data in security markets; 2) the accessibility to large computational power promoted by technology advances.

High frequency data have some unique charateristics which do not appear in lower frequencies. They are good source of information to research the trading process and market microstructures. HFT strategies try to identify those microstructures and profit from their observations. In this chapter, we introduce a few basic HFT examples. Before that, let's describe some emprical characteristics of high frequency transaction data. (Note: those are intercepted from Book [1]).

1. *Unequally spaced time intervals* - the time durations between trades are not identical, and they provide useful info about trading intensitiy. E.g., non-synchronous trading results in a non-exploitable fake signal.
2. *Descrete-valued prices* - the price change in one transaction for a security only occurs in multiple of tick size. From 2001/1/29, all NYSE and AMEX stocks started to trade in decimals. E.g., bid-ask bound results in a non-exploitable fake signal.
3. *Existence of a daily periodic or diurnal pattern* - for instance, on the NYSE, heavier trading at the beginning and closing of the trading hours and thinner during lunch hours.
4. *Multiple transactions within a single second* - transactions could occur in milliseconds.

Example A: The Most Profitable Pattern in A Search Space

Say we have a set of bid-ask price tick data for a stock in a trading session: $\{(b_t, a_t)\}_{t=1,\ldots,10^6}$, and we denote $\{p_t = \frac{b_t + a_t}{2}\}_t$ as the mid-price at time t. Note: here we assume the time difference between two consecutive ticks is the same, which is **totally wrong** in the real market, i.e., transactions occur with unequally spaced time intervals. Now we limit ourselves to trade patterns observed on previous two tick changes: $[\Delta p_{t-1}, \Delta p_t]$. There are four enumerated combinations, which include $[+,+], [+,-], [-,-], [-,+]$ if we assume the mid-price tick change has to be either positive or negative (note: this is not true since a majority of transactions are indeed without price change). These four combinations constitute the search space of four patterns, and we can build totally eight strategies around the four patterns, i.e., we can either buy or short the stock based on each of the observed patterns.

To make it more realistic, for each strategy, let's assume after we observe the corresponding pattern, we'll enter the position accordingly and hold it for a determined length of time (this is a model parameter) before exiting the position.

The most dangerous aspect when "reading" patterns solely from time series is the selection of spurious signals which exhibits strong back-tested (in sample) performance but poor live (out-of-sample) performance. There are different ways to avoid the overfitting risk, e.g., running tests on out-of-sample time series, and/or reducing model complexity level and number of parameters.

Now we need to decide which of the eight strategy would be implemented in the next trading session for the stock? In the following we introduce two methods to make this decision.

- *Back-testing the signal strength:* Similar to building traditional non-HFT investment strategies, we use a look-back time window for strategy selection and back-test the portfolio performance based on strategy momentum. For example, we run the portfolio by using the strategy which performes the best in the past three trading session. The back-test result sheds lights on the persistence of pattern profitability in history, and provides us clearance in whether to implement the signal out-of-samply. Note: the back-test result may show no profit associated with those pattern-based strategies.

- *Evaluating the existence of true alpha:* If there exists a systematic alpha from trading one of the patterns in the search space, then we'd better make sure it's not due to noise. To evaluate the profit of implementing one strategy is truely signficant, we compare the best possible profit level from implementing the same strategy on a randomly-generated return time serie. We can bootstrap to simulate a set of random return time series, i.e., by choosing from the actual return time series one by one with replacement to build a time series of the same length. If the profit of the strategy on actural return data is significant, it should be higher than 95% of the profits generated on those random time series when we set our confidence level to 95%. After we find a strategy with significant alpha, we can observe the duration of those significant market pattern based on historical data, and we then decide how to best captialize the findings.

In this example, the number of patterns in the search space is relatively small. It would require exponentially more computational power to caputure potentially profitable trading patterns as we enlarge the search space.

Example B: Pair-Trading Statistical Arbitrage

1. Regression-based price convergence signal

In this example, we again look at intra-day price data of a stock (or an ETF), and try to capture the mean reversion of the asset's price to its "fair price" which can be evaluated based on other highly correlated assets. For example, assume $\{y_t\}$ is the tick price time series for an asset, and $\{X_t\}$ is a multivariate price time series for other assets whose price moves we believe correlate with the price of the interested asset.

We can use weighted least squares (WLS) to estimate the following linear regression model:

$$y_t = \beta_0 + \vec{\beta} X_t + \epsilon_t,$$

where the coefficient parameters can be estimated based on WLS and we give more weight to more recent observations. At each time point, we can choose a static look-back window and based on observations in the time window to estimate the current fitted value $\overline{y_t}$. Then we buy (sell) the asset if the fitted price value is lower (higher) than its observed price level.

The above model has the following system parameters: the look-back time window, the weighting vector for WLS, the execution scheme for observed signals. The execution scheme typically outlines the specific price band for buy/sell, and for holding or exiting the exiting position, etc. The calibration of system parameters requires testing on model applicability to other asset pairs and model sensitivity to chosen parameters, etc.

2. Index ETF arbitrage opportunity

An index ETF is an investable vehichle which tracks the performance of a corresponding index. Arbitrage opportunity emerges if the ETF price level doesn't reflect its fair value as indicated by the index price moves. Let's denote $\{SPY_t\}$ and $\{SPX_t\}$ as the tick price time series for the ETF SPY and its benchmark index SPX (S&P500 Index). In this example, because the ETF we chose is very liquid, we don't have to worry much

about the price discrepancy due to liquidity issue. The market is efficient enough that we should observe little arbitrage opportunity on the close-price to close-price moves. However, in intra-day trading, i.e., when we observe ETF prices changes in transaction by transaction, on a high frequency basis, there may exist non-exploited arbitrage opportunities.

At time t, we caluate the average price ratios of ETF vs Index in the past h observations: $R_t = (\sum_{k=1}^{h} \frac{SPY_{t-h}}{SPX_{t-h}})/h$. We will compare the current ratio $R_t^* = \frac{SPY_t}{SPX_t}$ with the observed trailing moving average ratio, and we expect the two ratios should be close to each other. An arbitrage strategy can de deployed by identifying a threshold pair (c_l, c_h), where we buy the ETF if $R_t^* < c_l \cdot R_t$, and sell the ETF if $R_t^* > c_h \cdot R_t$. Again we need to specify when to exit the positions and choose proper system parameters.

Example C: Supply/Demand Signal From Limit-Order Book

Limit-order book of a stock offers the prices and quantitities traders are willing to buy and sell at each each time point. One can measure the supply/demand signal of a stock at any time point based on data from its limit-order book. There are different ways to quantify this signal. For example, we can define the signal as follows:

$$z_t = \frac{B(t) - A(t)}{B(t) + A(t)},$$

where $B(t)$ represents an aggregate demand measure based on bid data and $A(t)$ represents the supply measure from ask data; and a higher signal value indicates increasing demand compared to the previous snapshot time point. The detail function form for $B(t)$ and $A(t)$ varies but they should incorporate the price and quantity info. For example, we can define as follows: $B(t) = \sum_{k=1}^{h} r_{b,k} q_{b,k}$ and $A(t) = \sum_{k=1}^{h} r_{a,k} q_{a,k}$ where h is the levels of bid/ask orders we'll take into account, $r_{b,k}$ and $r_{a,k}$ resprents the distance to the best implementation price for each level of bid and ask orders respectively, and $q_{b,k}$ and $q_{a,k}$ are the quantities for those orders.

One can build a strategy around the signal z_t. We first need to decide the trading frequency, i.e., how often we check the limit-order book. And then buy/sell signal is a function of all the observed signal values up to each time point. The supply/demand-based strategies aim to frontrun other market participants before a price move occurs. A naive strategy is that if the signal value is above or below certain thresholds, we buy or sell the stock accordingly. More sophisticated strategies could exploit the technical patterns of the signal time series related to stock price moves. We can also build machine learning model where the dependent variable is the next-period price move and the explantory variables include the supply/demand signal histories.

Appendix

[1] Tsay, Ruey S. "Financial Time Series." *Wiley StatsRef: Statistics Reference Online* (2014): 1-23.

[2] Avellaneda, Marco, and Sasha Stoikov. "High-frequency trading in a limit order book." Quantitative Finance 8.3 (2008): 217-224.

[3] Avellaneda, Marco, Josh Reed, and Sasha Stoikov. "Forecasting prices from Level-I quotes in the presence of hidden liquidity." *Algorithmic Finance* 1.1 (2011): 35-43.

[4] Loveless, Jacob, Sasha Stoikov, and Rolf Waeber. "Online algorithms in high-frequency trading." *Communications of the ACM* 56.10 (2013): 50-56.