Machine Learning Engineer Nanodegree

Capstone Proposal

Application of Machine Learning Algorithms to Intraday Stock Trading Based on Demand Zones

Alexey Ganin
October 22nd, 2018

Introduction

In this project, I'd like to research applicability of Machine Learning methods to intraday stock market trading. Specifically, I'll focus on evaluating so-called "Demand Zones" in terms of potential profitability. I've built a system which tracks the S&P 500 stock data, detects the Demand Zones (time T_1), records the technical parameters, including the peak price (time T_2), and generates alerts when the price comes back to the demand zone (time T_3). At the time T_3, a trader (or a trading algorithm) has to make a decision on whether they should buy the stock. The purpose of my research is to analyze the performance of Machine Learning algorithms in terms of their ability to correctly predict the winning trades.

Domain Background

Supply and Demand are the key concepts of a market economy. Since market economy is based on exchange of goods and services for a value, for it to function there must be some goods and services to offer (supply) and people who are willing and able buy them (demand).

Trading financial instruments (Stocks, Forex, Futures, etc.) takes place in markets. In order to function, market needs sellers and buyers. The concepts of Supply and Demand Zones in trading are mostly concerned with spotting where buyers and sellers are sitting on trading charts (price ranges). However, usually retail traders do not have access to the order flow, so it is not possible to precisely spot the buy and sell orders.

Supply and Demand Zones method helps to identify price intervals with, presumably, high concentration of sell and buy orders. When the price comes back to one of such areas, the exchange starts to fulfill those orders. If the total volume of orders is significant, the price will react by moving in an opposite direction. Using lagging Supply and Demand information, traders are making trading decisions based on historical data.

There is an important difference between classic Supply and Demand theory and Supply and Demand that applies to trading. In classic approach suppliers generally stay as suppliers in the process of exchange, however in trading we can't identify certain participants as sellers or buyers. All participants in trading can be buyers or sellers at any point of time.

I'll focus my research specifically on Demand Zones, which are the stock price interval with, presumably, high number of unfulfilled "buy" orders. When the price comes to such an interval, the exchange starts to fulfill the "buy" orders, which, in turn, causes the price growth. The nature of a demand zone is best described by a picture:

Figure 1. Demand Zone Example

Note that the price starts growing when it "touches" the demand zone.

Demand Zones' nature is very similar to the one of Support Levels'. The key difference – the Demand Zone has wider range of prices. There are several ways for determining the width of a Demand Zone.

There are two types of Demand Zones formations: [1] Rally-Base-Rally (RBR) and Drop-Base-Rally (DBR).

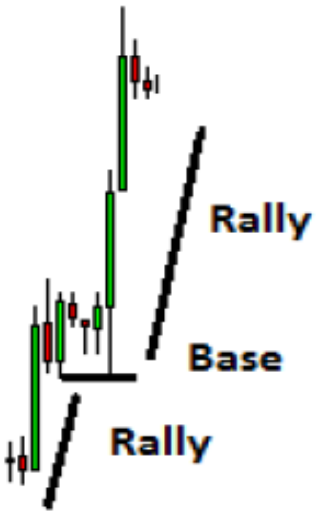

Figure 2. Rally-Base-Rally

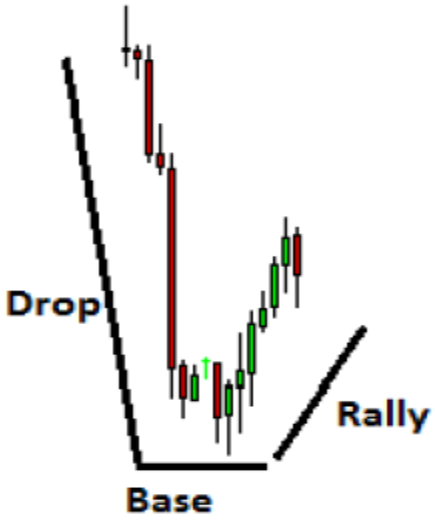

Figure 3. Drop-Base-Rally

In general, the probability of price growth decreases when the Demand Zone was "visited" several times. Some sources refer to number of visits as "freshness", other use the term "strength". When a zone was visited one or more times it is considered less fresh or less strong (See the Figure 4).

[1] Supply and Demand Zones - http://www.finanzaonline.com/forum/attachments/forex/2212736d1454155414-eur-usd-di-tutto-di-piu-149ma-ed-acegazettepriceaction-supplyanddemand-140117194309-phpapp01.pdf

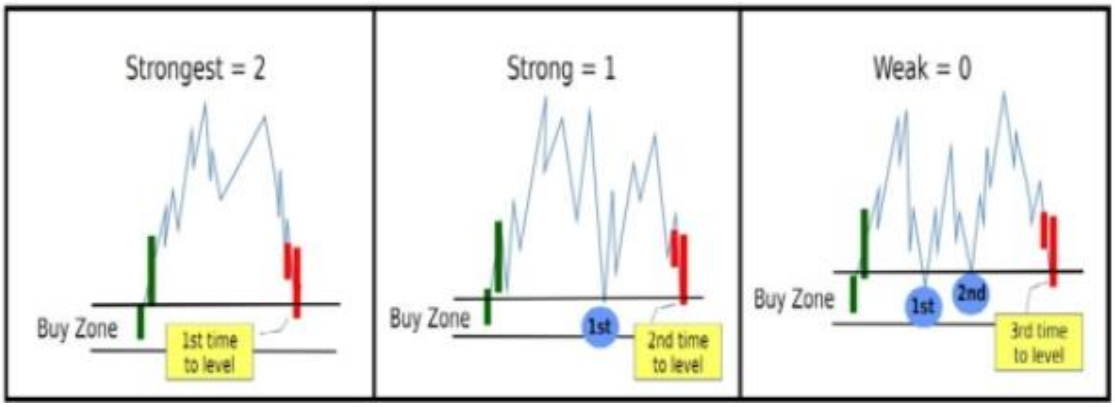

Figure 4. Strength/Freshness of a Demand Zone.

There is a similar concept of Supply Zones which are used as indicators of upcoming price decline.

More information on Supply and Demand Zones:

- https://www.investopedia.com/terms/z/zone-of-support.asp
- https://www.tradingacademy.com/lessons/article/two-things-matter-trading/
- http://www.finanzaonline.com/forum/attachments/forex/2212736d1454155414-eur-usd-di-tutto-di-piu-149ma-ed-acegazettepriceaction-supplyanddemand-140117194309-phpapp01.pdf
- https://www.colibritrader.com/what-are-supply-and-demand-zones-and-how-to-trade-with-them/
- https://hacked.com/trading-101-trading-supply-demand-zones/

Personal motivation. I was researching Supply and Demand Zones for about three years before staring the Udacity Machine Learning Nanodegree. I accumulated data and experimented with statistical analysis. Based on my research, I've created an automated system which monitors stock market in real time, recognizes Supply and Demand Zones, records market conditions, generates trading alerts (learn more at http://www.stocksbuyalerts.com/). This capstone project is a perfect opportunity to apply the knowledge I got during the course to the area I am passionate about and bring the quality of stock trading alerts generated by my system to the next level.

Problem Statement

The primary challenge of algorithmic trading is a high complexity and variability of the conditions which impact the stock price. Of course, Demand Zones, as any other trading

indicator, does not guarantee profitability. Additional challenge is the high level of noise on the short time intervals (e.g. one minute).

In my research, the algorithm's task will be to analyze the market data and determine the probability of a profitable trade at the point of a Demand Zone. The algorithms will define the entry points (buy-points) with the high probability of a short-term price growth. I will analyze and compare the performance of different machine learning algorithms for the specified task.

Datasets and Inputs

During the past three years, I was collecting stock market data related to Supply and Demand Zones for the stocks from S&P 500 list[2]. Major data providers I used: Interactive brokers[3], E-trade[4], Google Finance[5], Yahoo Finance[6]. In addition to the standard market data (timestamp, open, close, high, low, volume) in one-minute resolution, I've collected such parameters as:

- Bid price
- Bid size
- Ask price
- Ask size
- Day open
- Previous day close
- Day high (for the current point)
- Day low (for the current point)
- Last trade's price
- Total number of trades in a current day
- Total volume in a current day

All parameters above are gathered for with roughly 20 seconds intervals.

Sizes of the datasets:

- One-minute candles – about 191 million records
- Detailed stock data snapshots – about 423 million records

[2] List of S&P 500 stocks and their industries is taken from this resource: https://en.wikipedia.org/wiki/List_of_S%26P_500_companies
[3] Interactive Brokers - https://www.interactivebrokers.com/en/home.php
[4] E-Trade - https://us.etrade.com/home
[5] Google Finance - https://www.google.com/finance
[6] Yahoo Finance - https://finance.yahoo.com/

- My system recognized about 304 thousand Demand Zones of Drop-Base-Rally type during past three years.

I plan to analyze the following parameters of Demand Zones:

- growth – percentage of a price change between times T1 and T2 (refer to Terms and Definitions or Figure 1 for definition of T1 and T2)
- sma_dif – difference between price at time T3 and simple moving average (SMA) price for last 50 minutes.
- coefGT – ratio between growth and time interval (in minutes) between T_1 and T_3. It is calculated as:

$$coefGT = 10000 * \frac{growth}{(T3 - T1)}$$

- dif_high – percent difference between current day high and current price (%)
- dif_low – percent difference between current day low and current price (%)
- dif_open – price difference with current day open (%)
- dif_prev_close - price difference with previous day close (%)
- rsi – Relative Strength Index[7]
- macd - Moving average convergence divergence[8]
- volatility - intraday volatility calculated as standard deviation of closing prices for each minute.
- correlSP – correlation of current stock with S&P 500 index
- SP_SMA_dif – difference of current S&P 500 index value and 50-minute simple moving average (%)
- ind_SMA_dif – difference between current industry average value and its 50-minute simple moving average (%)

Here is the list of industries I will consider:

- Communication Services
- Consumer Discretionary
- Consumer Staples
- Energy
- Financials
- Health Care
- Industrials
- Information Technology
- Materials

[7] RSI definition - https://www.investopedia.com/terms/r/rsi.asp
[8] MACD definition - https://www.investopedia.com/terms/m/macd.asp

- Real Estate
- Telecommunications Services
- Utilities

Solution Statement

In simple words, each particular algorithm will need to analyze the stock price behavior, behavior of other stock prices in the same industry, and overall the situation on the market, in order to predict whether the trade is going to be profitable or not. I will use static target and stop-loss prices.

Since my goal is to analyze the performance of different Machine Learning algorithms in the field of stock trading, I will generate multiple solutions and then compare their metrics. For each particular algorithm the solution will be represented as a name of and algorithm, set of hyper parameters, the resulting model, and set of metrics described below.

Benchmark Model

The performance of each particular machine learning algorithm will be compared with performance of other algorithms. Other benchmark models will be: performance of S&P 500 index, and performance of a random decision making.

The benchmark model based on random decision making will function as follows: each time when the stock price approaches a demand zone, the algorithm will toss a virtual coin and, based on the outcome, decide whether it should enter a position. Performance will be measured as a sum of trade results for a particular period.

Evaluation Metrics

I will use two types of metrics: technical metrics of machine learning algorithms and profitability related metrics.

Technical metrics will include:

- Classification accuracy – how accurately an algorithm can separate winning trades form losing trades
- Logarithmic loss
- Confusion matrix
- F-score
- Mean absolute error

Profitability related metrics will be calculated based on performance of a trading strategy based on a particular ML algorithm over specific period of time (on previously unseen data). I'll use such metrics as:

- Sharpe ratio
- Total return
- Number of trades
- Number of winning trades
- Number of losing trades
- Average trade duration
- Average number of trades per day
- Maximum drawdown over the test period
- Maximum intraday gain
- Maximum Intraday loss
- **Monte Carlo simulation** results:
    - Number of scenarios
    - Summary chart of returns distribution
    - Average return ($,%)
    - Average max drawdown ($,%)
    - Return to drawdown ratio
    - Number of times account was ruined (minimum balance achieved)
    - "with probability of X% your strategy's return will be at least Y%"

Monte-Carlo simulation[9]

Based on historical performance of your trading strategy, you can build an equity curve. It's a great way to visualize your strategy's performance. But what if the order of trades was different? How would it impact your equity curve? Could your drawdown be more severe? These are the questions Monte Carlo simulation can answer.

In a simple form, Monte Carlo simulation can be explained as following: First, get a number of little pieces of paper, one for each trade in your strategy. Then, write down one trade result on each piece of paper and put all the pieces in a hat. Choose one piece randomly. That is your first trade. Record it, adding it to your initial equity, and then put the piece of paper back in the hat. Then, pick another piece of paper, record its value, and add it to the existing equity curve you are building. If you repeat this for a number of trades, you'll get a possible equity curve. If you do the whole analysis many, many

[9] Source: my website http://stockstrategytest.com/

times, you'll have a family of equity curves. Each of them will represent a possible scenario or sequence of your trades.

Using the family of possible curves, you can get statistics about your trading system. These statistics can help you evaluate a strategy, determine a position sizing approach, and give you realistic scenarios for what you might face if you actually trade the strategy live. Of course, this all assumes that the historically derived trades will be the same as the trades in the future. If your historical trades are based on flawed development, future results will be garbage.

Monte Carlo simulation is particularly helpful in estimating the maximum peak-to-valley drawdown. To the extent that drawdown is a useful measure of risk, improving the calculation of the drawdown will make it possible to better evaluate a trading system or method. Although we can't predict how the market will differ tomorrow from what we've seen in the past, we do know it will be different. If we calculate the maximum drawdown based on the historical sequence of trades, we're basing our calculations on a sequence of trades we know won't be repeated exactly. Even if the distribution of trades (in the statistical sense) is the same in the future, the sequence of those trades is largely a matter of chance.

Obviously, there are some potentially serious drawbacks to this analysis. First, the analysis assumes that the trades in your performance report are the only possible trades that can happen. This is obviously false, since when you start trading live, any result is possible for a particular trade. But, if the distribution (overall mean and standard deviation) of the trades is accurate, then the Monte Carlo approach can yield meaningful results.

A second drawback is that this analysis assumes that each trade is independent of the previous trade, a condition commonly referred to as serial or auto correlation. For most trading strategies, this is not an issue. However, if you have a strategy in which the trade results depend on each other, simple Monte Carlo analysis is not appropriate.

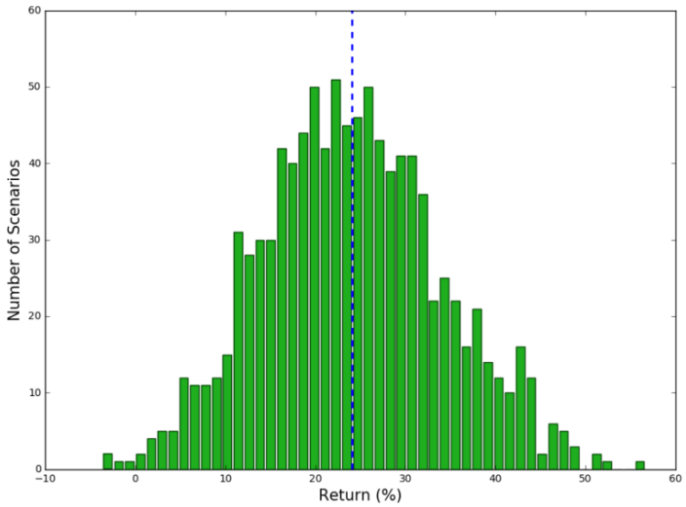

Figure 5. Monte Carlo simulation

Assuming you can live with the drawbacks listed, Monte Carlo can help you answer the following questions:

- What is my risk of ruin for a given account size?
- What are the chances of my system's having a maximum drawdown of X percent?
- What kind of annual return can I expect from this trading system?
- Is the risk I am taking to trade this strategy appropriate for the return I am receiving?

Project Design

I will start with data exploration. I'll create some visualizations in order to understand how each feature is related to the others. I will observe a statistical description of the dataset and consider the relevance of each feature.

As a next step, I'll do data pre-processing, which is a critical step in assuring that results obtained from analysis are significant and meaningful. Some features' data may not be normally distributed, especially if the mean and median vary significantly (indicating a large skew). I may need to apply a non-linear scaling. One way to achieve this scaling is by using a Box-Cox test, which calculates the best power transformation of the data that reduces skewness. A simpler approach which can work in most cases would be applying the natural logarithm. Also, as part of data pre-processing, I'll remove outliers.

Initial feature space will contain 13-18 dimensions. In order to increase the efficiency of the algorithms, I may need to apply dimensionality reduction methods such as Principal Component Analysis (PCA). When using principal component analysis, one of the main

goals is to reduce the dimensionality of the data — in effect, reducing the complexity of the problem. Dimensionality reduction comes at a cost: Fewer dimensions used implies less of the total variance in the data is being explained. Because of this, the cumulative explained variance ratio is extremely important for knowing how many dimensions are necessary for the problem.

I plan try applying the following ML algorithms:

- KNN
- Decision Trees
- SVM
- Naive Bayes
- AdaBoost
- Deep Neural Networks

I will do comparative analysis of the algorithms' performance as well as feature importance. I'll conclude by selecting the most efficient algorithm.

Scope of work

In Scope:

- I will consider prices around Demand Zones as potential entry points only.
- I will consider only Demand Zones of Drop-Base-Rally type.
- I will consider only Demand Zones which were not visited before ("freshness" = 0)
- I will set static target and stop-loss prices.

Out of scope:

- Assets other than S&P500 stocks (Forex, Futures, Options, etc.)
- Time intervals other than 1 minute
- Supply Zones

Optional areas of research (if I have time before the deadline):

- Applicability of Reinforcement Learning for defining optimal target prices when the order was placed. This will include training the agent, so it takes actions ("hold" or "sell") depending on the market situation in order to maximize the overall profitability. In this case the target and stop-loss prices won't be static anymore, but defined dynamically by the agent based on previous experience.