IS5006 Intelligent System Deployment
Group 03 - Final Project Report

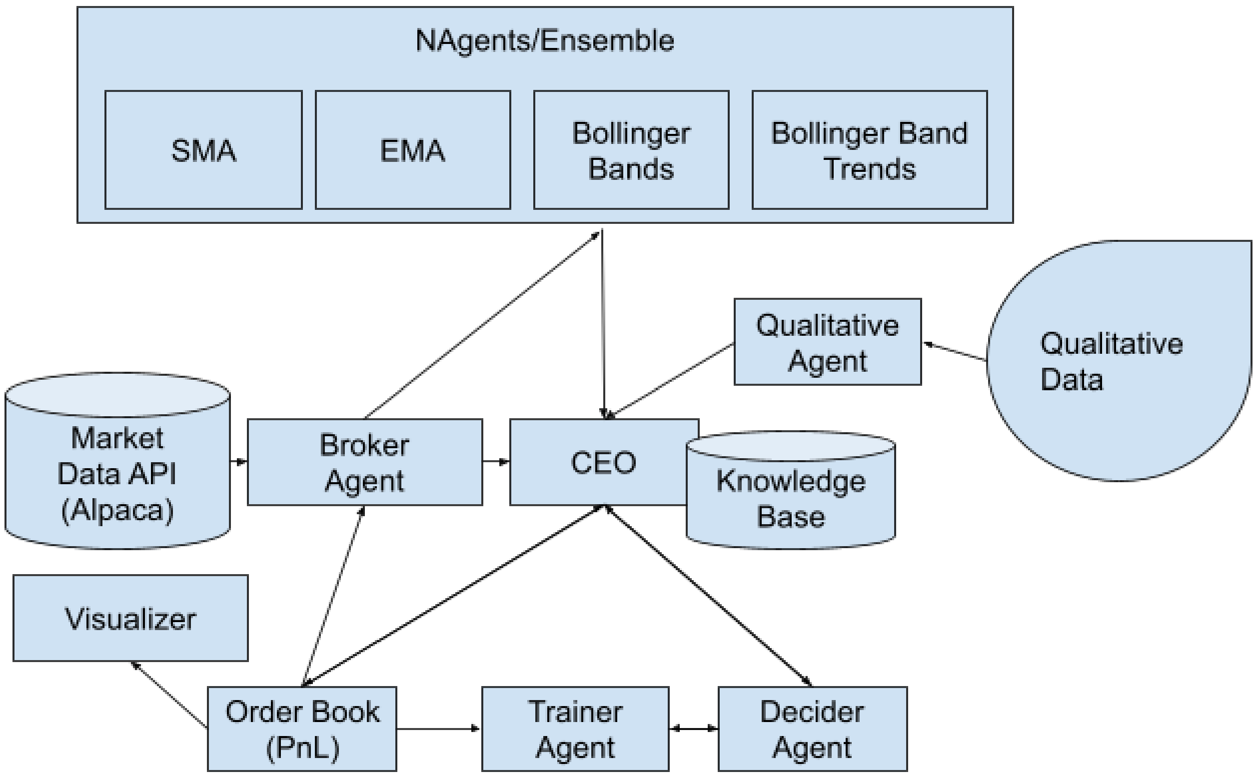
10th April 2022

Jesper Lim Jun Hao (A0248996B) e0934611@u.nus.edu
Pitchappan Pitchappan Ramaswamy (A0236575W) e0740930@u.nus.edu
Seow Kah Yong (A0032747E) e0383424@u.nus.edu
Zhu Yexin (A0083629W) e0696677@u.nus.edu

Table of contents

1. Our MAS Architecture

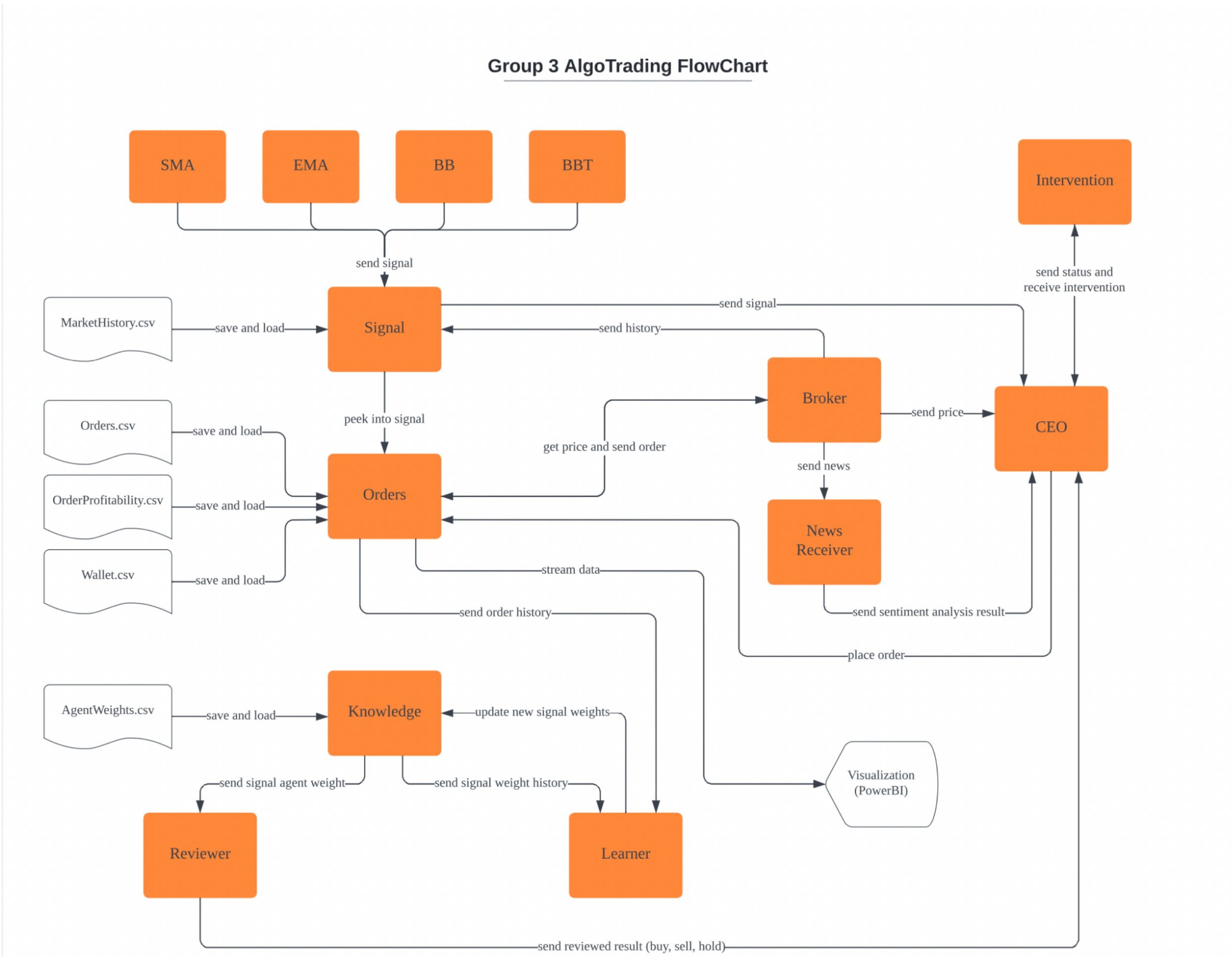
Here is a diagram that illustrates the architecture of our implementation:



The benefits of such an MAS implementation are similar to what Hopgood, A.A.(2005) has pointed out: good to solve complex problems and distributed problems.

2. Program Structure and Data Flow

Here is our program structure which consists of the different classes that make up our agents and the data flow between them. The arrows show a typical program flow within the system from generating signals out of historical data to training of the weights for the quantitative agents for the next iteration:



2.1 NAgents

The NAgents component (I.e. Signal) follows Tirea, M.,Tandau,I. & Negru,V.(2011)'s model where a technical analysis agent contains various technical indicators (I.e. an ensemble). The outputs of these indicators generate buy/hold/sell signals and are computed with trainable weights by the Decider agent (I.e. Reviewer).

Each indicator has three weights associated with it for each type of signal it generates (I.e. [type of indicator,buy weight,hold weight,sell weight]). This is because a specific type of indicator might not be good in predicting buys but it is very good in predicting sells. Separating weights for each of these possible actions provides us that information.

Lastly, we train these weights based on a perceptron design[1].

Here are the technical indicators that we are using:

2.1.1 SMA/EMA

As there are various sources on what SMA/EMA[2][3] is, we will not be covering the algorithm for our SMA/EMA agents. On an additional note, we have configured a fixed short window period of 20 and a fixed long window period of 50 for both these indicators as a result of our previous trials and that yield the most favorable result.

2.1.2 Bollinger Bands

We will not be covering the algorithm of our BB agent as there are various sources[4] on this. On an additional note, we have configured the window period of 20 for this indicator as our previous trials show the most promise for this value. We also use 2 standard deviations (STD) away from the moving average.

2.1.3 Bollinger Band Trends

Contrary to some sources[5] of BBT implementation, our implementation differs from it. Here are the details of our implementation:

1. Calculate the low and high bands of two BBs with different time windows (I.e. 20 and 50).
2. Take the difference between the lower bands of 20 and 50 BBT and higher bands of 20 and 50 BBT.
3. Take the difference of lower band difference and the higher band difference from step 2. and take a quotient of the moving average of the lower window. (I.e. lower band difference - higher band difference / SMA20)
4. If the result from step 3 is bigger than 0 and we do not have a buy signal previously, we will generate a buy signal. If it is equal to 0, we generate a hold signal. And if it is less than 0, we generate a sell signal if we have not done so earlier.

2.2 Qualitative Agent (News Receiver)

This agent basically checks for online news to get a feel on the current market sentiment. We are using Spacytextblob[6] for our sentiment analysis instead of twitter. It gives us two different

---

[1] Jason,B.(2016).How To Implement The Perceptron Algorithm From Scratch In Python, https://machinelearningmastery.com/implement-perceptron-algorithm-scratch-python/

[2] Adam,H. (2022). Simple Moving Average(SMA). https://www.investopedia.com/terms/s/sma.asp

[3] James.C. (2022). Exponential Moving Average (EMA). https://www.investopedia.com/terms/e/ema.asp

[4] Adam,H. (2022) Bollinger Band®. https://www.investopedia.com/terms/b/bollingerbands.asp

[5] Mitchell,C.(2022). Using Bollinger Bands to Gauge Trends, https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/

[6] https://spacy.io/universe/project/spacy-textblob

values: subjectivity and polarity which will be applied as antecedents for our fuzzy logic to decide the volume that we will trade in.

2.3 CEO and Intervention

The CEO is similar to the coordinator agent described in Tirea, M; Tandau, I; Negru, V (2011)'s model.

Here are its responsibilities:

1. Getting signal importance from the Reviewer which takes in inputs from the ensemble.
2. Using case-based reasoning (CBR) to determine if a given action is good.

   Our CBR score is computed as such:
   a. We consolidate all the profitability reports from the Orders class for all the quantitative agents.
   b. We count the number of times the orders make a profit P, the number of times the orders make a loss L and the number of times the orders do not make a difference N.
   c. We derive the CBR score using the formula (P - L) / (P + L + N)

   The CBR score is used mainly as a multiplier to determine the volume size.

3. Decide the buy/sell order quantity and price via the CBR score and risk appetite.
4. It sends details of the order to the PnL agent (Orders), which is sent to the broker agent (I.e. Broker) to place the order.
5. Monitors News outlook of the market and warns the user or stops trades if the market outlook reaches extremes
6. It also takes in profit/loss reports from the PnL agent (Orders) and marks the times when these events happen.
7. Gets the balance from the Broker agent.
8. Allows the user to pause its execution (kill switch) via Intervention
9. Allows the user to change its risk level (I.e. level 1-6). The CEO's risk level determines its buy/sell/hold behavior.

2.4 Broker Agent

The broker agent serves as a broker that interfaces with the market data API for orders and getting historical data, which gets sent to the Signals class. It also reports the status of the order and the price that the order is filled. We are using the Alpaca API[7] for this and the API allows us to send news to the News Reviewer as well. Lastly, it sends/updates the balance to the CEO.

[7] https://alpaca.markets/

2.5 Decider Agent (Reviewer)

The decider agent takes the signals from the CEO and decides on the importance of the signal based on trained weight for each quantitative agent (QA) and returns that importance back to the CEO for further deliberation.

2.6 Trainer Agent (Learner and Knowledge)

The trainer agent (Learner) takes the current weights assigned (Knowledge) to the decider agent (Reviewer) and based on the profitability of the previous trades (I.e. performance measure), adjusts the weights of the QAs in the Knowledge class. We used delta rule training on a single perceptron to train the weight for each QA. The delta rule training function that we used is a linear function current weight * (1+(difference in profit/loss / (buy/sell price) * 10)).

2.7 PnL Agent (Orders)

The order book (agent) has various roles:

1. Keeps track of the details of all orders by adding and updating them (E.g. status of the order, filled price)
2. Relays stop loss/take profit/buy/sell orders to the broker agent
3. Updates the wallet by communicating with the broker agent
4. Keeps track of the orders' profitability for performance measure and training of weights

3. Strengths and Weaknesses

Strengths

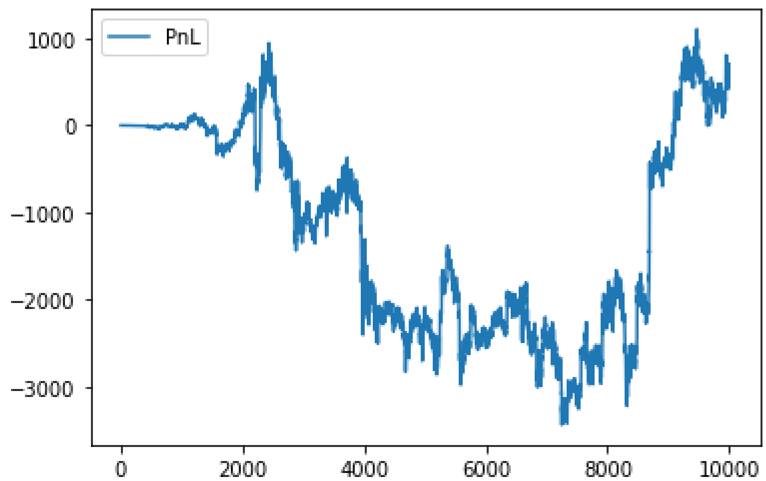
1. Does generate profits sometimes.
2. It is able to handle spikes in price action.

Weaknesses

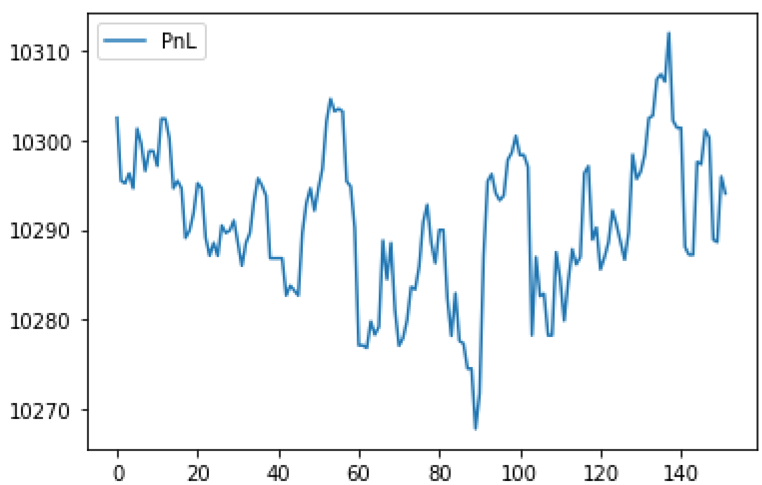
1. Too simplistic for the market.
2. Limit orders can just be left hanging in Alpaca if the system is shut down at a bad time.

4. Results

Here are the results from our back-testing, taking data for a period of 1 week (from Feb 9-15 2022). We made a comeback after making a loss.



Here are the results of our live-trading, taking data from April 9th 1 PM to 5 PM:



5. Bonus Implementations

- Implemented new signals such as Bollinger Bands and Bollinger Band Trends
- Proposed an interesting method to judge orders (using market trend) instead of pure Profits and Losses
- Applied Fuzzy Logic on News sources to judge overall market conditions for trading
- Added Human Intervention Methods to control the algorithm while it is live

6. Future Direction[8]

- Implement advanced signals from:
  - MACD
  - ML methods
- Use Fuzzy logic, News, current PnL to dynamically change risk value.
- Explore better order profitability.
- Explore more advanced CBR techniques.
- Better code structure to directly load history from Alpaca.

---

[8] The details for these improvements can also be found in the source code IS5006_Group3_Final_Live_Code_submission.ipynb.

References

Hopgood, Adrian. (2005). The State of Artificial Intelligence. Advances in Computers.

Bratman M.E., Israel D.J., Pollack M.E., “Plans and resource-bounded practical reasoning”, *Computational Intelligence 4* (1988) 349–355.

Brooks R.A., “Intelligence without reason”, in: *12th International Joint Conference on Artificial Intelligence (IJCAI’91)*, Sydney, 1991, pp. 569–595.

Tirea, M.,Tandau,I. & Negru,V.(2011). Multi-agent Stock Trading Algorithm Model, *2011 IEEE*

Jason,B. (2016). How To Implement The Perceptron Algorithm From Scratch In Python., https://machinelearningmastery.com/implement-perceptron-algorithm-scratch-python/

Hayes,A. (2022). Simple Moving Average(SMA). https://www.investopedia.com/terms/s/sma.asp

Chen,J. (2022). Exponential Moving Average (EMA).
https://www.investopedia.com/terms/e/ema.asp

Hayes,A. (2022) Bollinger Band®. https://www.investopedia.com/terms/b/bollingerbands.asp

Mitchell,C.(2022). Using Bollinger Bands to Gauge Trends,
https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/

Samip,T. (2020) ,Time Series Analysis — A quick tour of fbProphet,
https://medium.com/analytics-vidhya/time-series-analysis-a-quick-tour-of-fbprophet-cbbfbffdf9d8