MACHINE LEARNING FOR TRADING

Diana Yau

PROBLEM

- Given an in-sample period of a particular stock, how can we determine whether to buy, hold, or sell on an out-of-sample period using reinforcement-based learning?
  - In-sample period: January 1, 2008 to December 31, 2009.
  - Out-of-sample period: January 1, 2010 to December 31, 2011.

SOLUTION

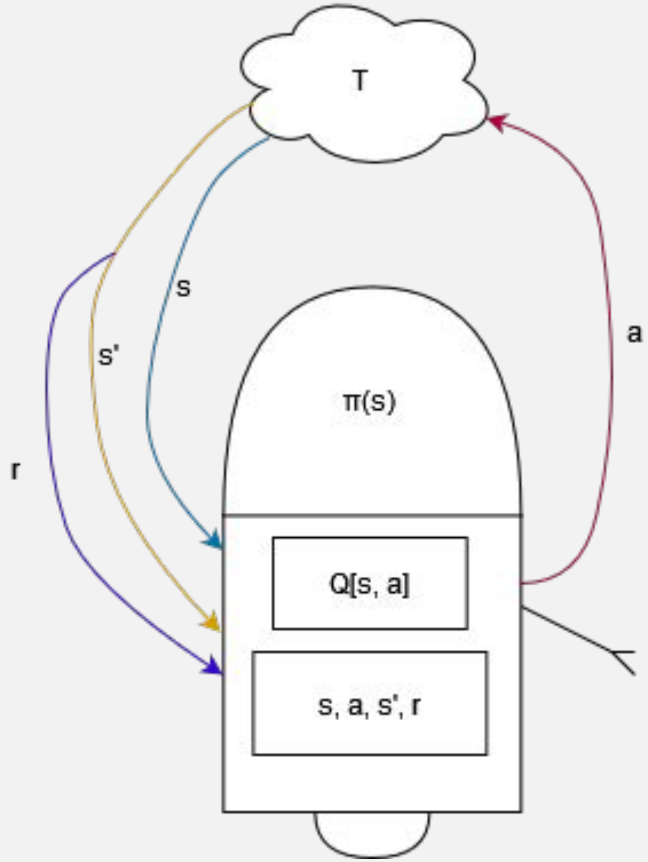
- Language: Python
  - NumPy and Pandas available as open-source.
  - Easy to access data.
- Q-Learning where Q[s,a] is a 2D table stores the consequent rewards where ‘s’ is the state and ‘a’ is the action.
  - Using combined and normalized indicators as states, query the table to get the new Qvalue
- Return a trades DataFrame (a table where the dates listed in the out-of-sample period correspond to a state).

IMPLEMENTATION – INDICATORS

- 3 indicators: Price/SMA, Bollinger Bands® %B, Momentum
  - Using indicator values of: Buy: 2, Hold: 1, Sell: 0.
  - Gives the maximum number of states as 222 and minimum as 000 with 3 actions.
  - For any day where there are 2 or more indicator values to buy/sell, consider there is a signal.
  - Consider having a window size of 10 days.

IMPLEMENTATION - QLEARNING

- Q-learning based strategy using a Q-Learner.
  - Not greedy because it takes the action at the current time.
- Policy $\pi(s) = argmax(Q[s,a])$
- Q[s,a] is a 2D table storing all known states and actions.
- A tuple of the state(s), current action(a), new state (s’), and reward(r) is passed iteratively.



IMPLEMENTATION – QLEARNING (CONT.)

- $$\underbrace{Q'[s,a]}_{\text{New Q value}} = \underbrace{(1-\alpha)Q[s,a]}_{\text{old value}} + \alpha(\underbrace{r + \underbrace{\lambda * Q[s', argmax(Q[s',a'])]}_{\text{discounted reward for all future actions}}}_{\text{new best estimate}})$$

α: *learning rate* (0 *to* 1), *typically* 0.2
λ: *discount rate* (0 *to* 1)
r: *immediate reward*

IMPLEMENTATION – TRAINING DATA

```
while not converged:
    x = indicators
    querySetState(x)
        for each day:
            r = calculate reward
            a = query(x,r)
            add action to dataframe of trades
            x = new state
    check if converged
```

CHALLENGES – FEATURE SELECTION

- Which indicators to use? What is a good number of indicators?
  - Price/SMA (current price/simple mean average)
  - Price/EMA (current price/exponential moving average)
  - Bollinger Bands® %B ([current price-lower bound]/[upper bound-lower bound])
    - Bollinger Bands® (SMA ± 2 Standard Deviations)
  - Momentum (current price / previous price by a specific window of days ago)
  - %Price (difference between two moving averages)

- Lagging indicators
  - Backfilled data by the window size

CHALLENGES (CONT.)

- How can the indicator be interpreted as buy or sell signals?
  - Suggestions: Buy: 1, Hold: 0, Sell: -1 or Buy: 2, Hold: 1, Sell: 0
- How can a combination of these indicators tell us what action to take?
  - How many indicators need to be validated to count the day as a buy/sell signal?
  - Considering the indicators as states?
- Trial and error: adjusting the learning rate and discount rate to maximize more trades
  - More trades = more potential to maximize holdings

FUTURE WORK

- Using a set cash value, compare the cumulative returns, mean and standard deviation of daily returns
  - Finding the maximum reward output by adjusting the alpha (learning rate) and lambda (discount rate).
- Comparing with a benchmark (S&P 500) and a classification-based learner, which can be a Random Forest learner.
  - Things to consider: Setting leaf size to avoid overfitting in-sample period
- Increasing the number of indicators to solidify position (e.g., RSI – relative strength index, stochastic indicator, etc.)
  - Different trading setups: Stops, trailing stops, stop-loss
- Wider in-sample period with considerations of special days (e.g., stock split forward/backward, fed rates hike, etc.)

Q&A