Option Pricing with Reinforcement Learning

Di Yu Tang, Randall Li
University of California, Santa Cruz
dtang10, rhli@ucsc.edu

Baskin Engineering UC SANTA CRUZ

CMPS 140 Artificial Intelligence 2019, UCSC.

Introduction

Motivation

Stock derivatives such as options allow for highly diversified trading strategies that will satisfy a wide spectrum of investment goals. Stock options is a classic hedging instrument for investment bankers where large capital is required. To the average non-commercial trader, options provide good cost efficiency and deliver high percentage returns due to the enormous amount of leverage exists for some contracts. The pricing model for stock options is highly sophisticated. Even when given historical option pricing data, pinpoint the exact trading strategy to maximize profit can be challenging. This optimization problem is an excellent use case for reinforcement learning. In a reinforcement learning environment, the agent will simulate day-to-day option trading with the historical pricing to explore rewards and transition between portfolio states.

Goals

- Developing an agent with reinforcement learning that will maximize profits from trading options
- Finding a optimal trading strategy by studying trends in the agent's transaction history

Options Review

An option contract offers the buyer the right but not the obligation to buy (call option) or sell (put option) of the underlying asset at an agreed-upon price (strike price) on a specific date (expiration).
Each contract carries a cost (premium) that the buyer pays to the option writer. The premium will fluctuate depending on how far away the strike price is from the stock price (moneyness), volatility, and time until expiration.

A call option is considered "in-the-money" when the strike is below the current stock price, "at-the-money" when strike is close to stock price, and "out-of-the-money" when strike is above the stock price. All of this is opposite for for put options. A contract is considered exercised when the underlying stock is bought or sold at the strike price. In the real world however, vast majority of the contracts will be sold for higher profitability instead of being exercised. A large capital is required to properly exercise contracts.

Table 1 is a simplified representation of what a typical option chain looks like. A heavily traded option gets around fifty to a hundred available strikes. Each strike will have a call side and a put side with bid and ask prices for each of them. Each expiration date has its own table. SPY options have more than twenty available expiration dates which is more than the average stock since SPY has weekly options. Therefore, a year of SPY options pricing data will contain more than a million entries.

Call		Expires on JUN 21, 2019	Put	
BID	ASK	STRIKE	BID	ASK
10.26	10.34	**276**	6.41	6.44
9.57	9.64	**277**	6.74	6.76
8.89	8.96	**278**	7.08	7.10
8.23	8.31	**279**	7.44	7.46
7.59	7.66	**280**	7.82	7.84
6.97	7.04	**281**	8.21	8.24
6.37	6.44	**282**	8.63	8.67

Table 1. Simplified option chain for SPY trading at 279.71 as of March 12 2019. An actual option chain will contain many more strike prices as well as important metrics such as volume, implied volatility, Theta, Delta, Gamma, etc.

Data

- The dataset was purchased by our TA Molly Zhang through historicaloptiondata.com and was generously provided to us
- The data contains historical SPY ETF option prices from 2005 to 2017. The entire dataset contains 15 million entries. We only utilized pricing data from January 2016 and the columns "Date", "Underling Price", "Option Type", "Expiration Date", "Strike", and "Ask".
- The dataset is further reduced down to January 4 to February 19 of 2016 containing only the options that costs more than $1 and expires within those months. In the end, we have 14,231 rows of data.
- The SPY ETF replicates the performance of the S&P 500 index. SPY was chosen for our dataset due to it's low volatility. Because SPY option has the highest volume, so spread between ask and bid is always within a few cents. Therefore we only use the ask price for trading and reduce the dataset.

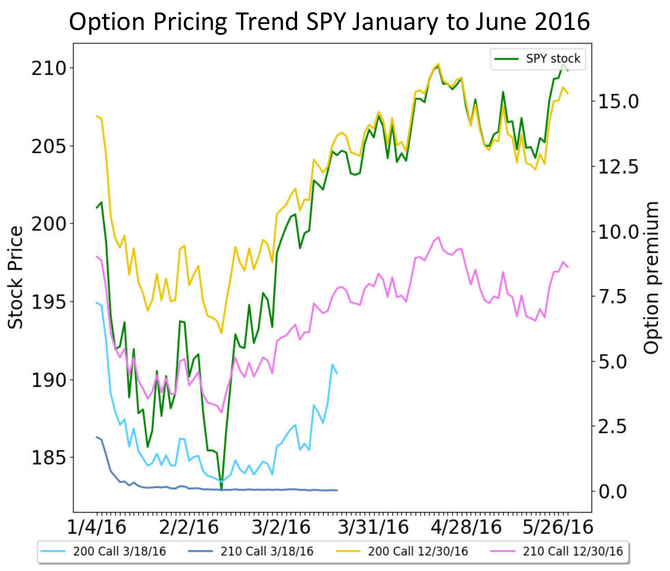


Figure 1. Moneyness and days until expiration plays a major role in how option pricing fluctuates. This figure is a comparison in how the premium of options with different strike and expiration dates differs over time and stock prices change. The 3/18/2016 options were cut off because that is when they expire. There are several option pricing properties that we need to consider in order to understand how an optimal strategy is derived:

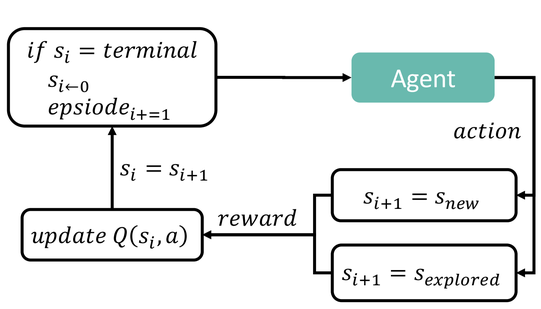
- Options with different strike prices but with the same expiration date will converge if they are both out of the money
- Premium becomes more volatile as the option is getting closer to expiration date
- Options further out-of-the-money will also be more volatile
- Options further in-the-money follows the stock price more closely
- Option premium decays after it looses its time value. As expiration day approaches, the decay accelerates.

Methodology

Algorithm

Q-Learning is a model-free reinforcement learning algorithm with the notion that the agent explores all possibilities of state and action pairs and estimate long term rewards by applying an action in that state-action pair. The learning process involves updating the Q-value for each state-action pair until Q-values converge. Q-Learning was chosen because rewards and transition functions are unknown and we want the agent to simulate explore rewards through day-to-day trading.

- **State** – state consists of date and a list of position currently holding. State starts at January 4, 2016 during each episode. After a transition, a new state may be added to the state space.
- **Action** – Each state is allowed to buy all available strikes given the restrictions below. After each transition, a "sell" action may be added to the action space of the next state.
  - Limited to buying 1 contract per day and able sell any amount of contracts
  - May not sell contracts that was bought on the same day
  - May not write contracts i.e. can only sell contracts that it's currently holding
  - May not exercise contracts for stocks
- **Transition** – After an action is taking, date is incremented and a new list of current positions are the next state's parameters. If the next state is new, it is added to the state space. If the calendar runs, then a new episode begins with starting state at January 4, 2016 holding no positions.
- **Reward** – Calculated profit in dollars from selling currently held options. If an option expires, the negative reward is the premium paid for the expire option.



Q-value update

$$Q(s_i, a) = (1 - a)Q(s_i, a) + a\left[r + \gamma(\max_a Q(s_{i+1}, \forall A)\right]$$

ε - greedy exploration

$$\varepsilon = 1/\sqrt{epsiode_i} \qquad x = random(0, 1)$$

$$action\ a = \begin{cases} random\ a, & x < \varepsilon \\ \max_a Q(s_{i+1}, \forall A) & x \geq \varepsilon \end{cases}$$

Parameters

- Learning rate $a = 0.5$ – For nearly all runs with low a, Q-values did not converge very well. With an high a closer to 1, the agent learned very different strategies during each run. We found that 0.5 gave the most consistent results.
- Discount rate $\gamma = 0.5$ – Having a lower discount rate makes agent more optimal, but takes more episodes to learn. Lower discount rate also implies that the agent will be greedier and prioritize immediate rewards.
- ε - greedy exploration was to enable the agent to occasionally choose a random action instead of one maximizes Q-value. This prevents overfitting and improves optimality. As episode increases, emphasis should be placed on exploitation. The ε - greedy function below allows for a gradual ε reduction.

Results and Evaluation

After around 30,000 episodes with the aforementioned Q-value update parameters and restrictions, the agent has derived an active or in other words greedy trading strategy that focuses on securing immediate profit. By evaluating figure 2 and the transaction history, we can conclude that these are optimal strategies:

- **Buy deep in-the-money options during volatile periods** – To adapt to the restriction of buying 1 option contract per day and that the reward function is based on dollar amount profits instead of percentage, the agent opted to trade deep in-the-money options that carries significantly more premium and a higher dollar return proposition.
- **Buy lowest premium options during flat trends** – The agent's available action every day is to either buy or sell options. During flat trends, agent has learned to buy the cheapest contract to minimize loss from time value decay.
- **Buy options with closer expiration dates** – Options closer to the expiration date have higher pricing fluctuation
- **Not holding options** – The agent has learned that selling profitable contracts as soon as possible and opening another trade is where profitable potential higher is a better strategy than holding long term contracts.

Figure 2. A simulation of the optimal trading strategy was ran and the agent was able to achieve $4991 profit from 1/4/2016 to 2/19/2016. Notice some profit dips during days when SPY is trading flat. Much higher profit is possible if the agent was not limited to single quantity trades. This however brings much greater complexity into this problem.

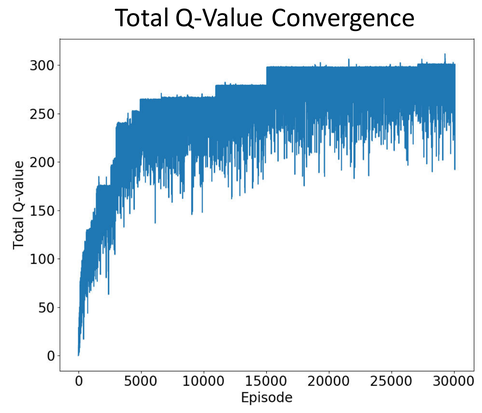


Figure 3. Convergence is consider reached when the difference between current Q-values for all states and the previous Q-values is less than 0.000001. At episode 10,000, ε is 0.01 based on our ε - greedy function, which means that the agent still has 1% chance of exploration. This is shown by the sharp dips in total Q-value for some episodes

Date	Trade	Q-Value	Profit	Date	Trade	Q-Value	Profit	Date	Trade	Q-Value	Profit	Date	Trade	Q-Value	Profit
1/4	buy 1/8/16 198.5 put	64.74	$-	1/14	buy 1/22/16 190.5 put	97.07	$1,620	1/27	buy 1/29/16 176.5 put	128.50	$2,662	2/9	sell 2/12/16 205.5 put	60.73	$4,041
1/5	buy 1/8/16 199 put	129.47	$-	1/15	sell 1/22/16 190.5 put	194.14	$1,790	1/28	buy 2/5/16 176.5 call	256.99	$2,662	2/9	buy 2/12/16 205.5 put	55.45	$4,041
1/6	sell 1/8/16 199 put	258.95	$70	1/15	buy 1/15/16 187 put	48.28	$1,790	1/29	sell 2/5/16 176.5 call	513.98	$3,146	2/10	sell 2/12/16 205.5 put	110.91	$4,080
1/6	buy 1/8/16 203 put	377.89	$70	1/19	buy 1/22/16 197 put	96.56	$1,773	1/29	buy 1/29/16 177 put	59.97	$3,146	2/10	buy 2/19/16 208 put	143.82	$4,080
1/7	sell 1/8/16 203 put	755.79	$595	1/20	sell 1/22/16 197 put	227.11	$1,974	2/1	buy 2/5/16 193 put	119.94	$3,143	2/11	sell 2/19/16 208 put	287.63	$4,279
1/7	sell 1/8/16 198.5 put	461.58	$1,001	1/20	buy 2/12/16 191.5 call	48.22	$1,974	2/2	sell 2/5/16 193 put	245.88	$3,352	2/11	buy 2/19/16 167 call	177.27	$4,279
1/7	buy 1/29/16 205 put	111.15	$1,001	1/21	sell 2/12/16 191.5 call	96.45	$1,992	2/2	buy 2/12/16 174 call	71.75	$3,352	2/12	sell 2/19/16 167 call	354.53	$4,590
1/8	sell 1/29/16 205 put	222.31	$1,207	1/21	buy 2/5/16 174 call	156.89	$1,992	2/3	sell 2/12/16 174 call	143.51	$3,444	2/12	buy 2/12/16 169 put	87.06	$4,590
1/8	buy 1/8/16 193 call	32.61	$1,207	1/22	sell 2/5/16 174 call	313.79	$2,273	2/4	buy 2/5/16 172 put	103.01	$3,444	2/16	buy 2/19/16 177.5 call	174.13	$4,589
1/11	buy 2/12/16 197.5 put	65.22	$1,206	1/22	buy 1/22/16 182.5 put	65.58	$2,273	2/4	buy 2/12/16 205.5 put	206.02	$3,444	2/17	sell 2/19/16 177.5 call	350.25	$4,922
1/12	buy 1/15/16 186 put	132.44	$1,206	1/25	buy 2/19/16 169 call	131.16	$2,272	2/5	sell 2/12/16 205.5 put	412.05	$3,800	2/17	buy 2/19/16 205 put	34.50	$4,922
1/13	sell 2/12/16 197.5 put	264.88	$1,405	1/26	sell 2/19/16 169 call	264.31	$2,473	2/5	buy 2/12/16 204 put	112.09	$3,800	2/18	sell 2/19/16 205 put	69.00	$4,991
1/13	buy 2/12/16 183 call	131.77	$1,405	1/26	buy 2/19/16 205 put	126.62	$2,473	2/8	sell 2/12/16 204 put	224.18	$4,008	2/18	buy 2/19/16 166 put	0.00	$4,991
1/14	sell 2/12/16 183 call	263.53	$1,620	1/27	sell 2/19/16 205 put	253.25	$2,662	2/8	buy 2/12/16 205.5 put	30.36	$4,008				

Conclusion

- With reinforcement learning methods such as Q-learning, we are able to develop an agent that learns the optimal option trading strategy with given parameters and restrictions. We expected similar learned trading behaviors would be adopted in order to maximize profit, but it is difficult to verify whether the transactions are truly optimal.
- Limited computation power was a major setback since the state-action space becomes exponentially larger as more historical data was used. Therefore, episode iterations often hit a predefined limit before the agent has truly maximized profits, but figure 3 shows that the agent is close to optimal.
- Selected the learning rate, discount rate, and ε function was time consuming task. The effectiveness of these parameters seemed to heavily rely of the size of the dataset. Each combination of possible value for these three parameters produces significantly different trade routes, but they all follow closely to the optimal strategy.
- Filtering pricing data was an unexpectedly difficult task especially when we tried to remove low cost contracts. The data still has to follow the properties of option pricing and we frequently broke the simulation by not including contracts that are supposed to exist.

Reference

Kansal, S., & Martin, B. (n.d.). Reinforcement Q-Learning from Scratch in Python with OpenAI Gym. Retrieved March 12, 2019, from https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/
Huang, S. (2018, January 12). Introduction to Various Reinforcement Learning Algorithms. Part I (Q-Learning, SARSA, DQN, DDPG). Retrieved March 14, 2019, from https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287
Singh, G. (2018, January 23). Deep Reinforcement Learning for Algorithmic Trading. Retrieved March 14, 2019, from https://medium.com/@gaurav1086/machine-learning-for-algorithmic-trading-f79201c8bac6