CMPS 140 Final Report: SPY Stock Options Pricing with Reinforcement Learning

Randall Li (rhli@ucsc.edu)
Di Yu Tang (dtang10@ucsc.edu)

Abstract: Stock derivatives allow for a diversified spread of investment opportunities, particularly stock options. SPY stock options are a popular choice for both investment bankers and average non-commercial traders, due to their low volatility, volume, and large historical dataset. The problem of maximizing profits given a time period is a thought-provoking challenge that can be solved by training an agent with historical data through Q-learning, a form of reinforcement learning. By studying trends in the agent's transaction history, we can also reaffirm some of the option pricing mechanics. The resulting optimal trading strategy that the agent learned was for the most part within our expectations in the case of hindsight trading. However, some actions were surprising but justifiable upon deeper analysis of the agent's environment.

Keywords: S&P 500, SPY, SPY stock options, options pricing, Q-Learning, reinforcement learning,

1. Introduction

1.1 Motivation

Stock derivatives such as options allow for highly diversified trading strategies that will satisfy a wide spectrum of investment goals. Stock options are 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.

1.2 Goals

- Developing an agent with reinforcement learning that will maximize profits from trading options with historical pricing data.
- Finding an optimal trading strategy and understand the agent's learned actions by studying trends in the agent's transaction history.

2. Theoretical Framework

Initially, reinforcement learning was chosen with a different goal of future option value prediction in mind. However, we discovered during the planning stage that there is a considerable disconnection between episodic learning and value prediction. For each of the agent's actions, a reward must be calculated. To calculate the reward, the future option pricing must be known. The agent will then reconsider its actions in the next episode to maximize future reward. This is equivalent to trading in hindsight and conflicts with the idea of future pricing prediction.

Not wanting to shy away from reinforcement learning, we updated our initial goals from value prediction to an optimization problem of maximizing profit by hindsight trading. This turns out to also be a challenging problem as described in the rest of this report. We were planning to run a simple value iteration on our data sample, but we realized that the state transition probabilities and rewards are not presented in the real scenario. In other words, we cannot present the agent with all possible rewards as that will be improbable given our computation limits. We moved on to a subset of reinforcement learning, Q-learning where the agent is able to explore states and rewards for each episode. Q-learning is not without its challenges of defining the states, actions, and optimizing the parameters for consistent results.

2. Data

2.1 Source

The dataset was obtained from our TA, Molly Zhang who purchased through historicaloptiondata.com and generously provided to us. The entire dataset contained 15 million entries of historical SPY ETF option prices from 2005 to 2017. The SPY ETF replicates the performance of the S&P 500 index. SPY was chosen as our dataset due to having weekly options. Weekly options provided more pricing data for a shorter period of time.

2.2 Data Reduction

We only utilized pricing data from January 4, 2016 to February 19, 2016 with the columns "Date", "Underling Price", "Option Type", "Expiration Date", "Strike", and "Ask". Since SPY options have the highest volume, the spread difference between the ask and bid price is always within a few cents. Thus, we reduced the dataset by not including the bid column and only used the asking price for trading. The dataset was further reduced down by containing only options that cost more than a dollar and expires within January 4, 2016 to February 19, 2016. In the end, we have 14,231 rows of data. Appendix Table 1 is how the dataset is represented in an option chain. Option pricing trend similar to appendix figure 1 can be observed if we graph the dataset.

Filtering the data was the first obstacle that we faced. Often times we excluded entries that would break the properties of option chain table, causing the agent to behave unexpectedly. For example, when we remove a contract that is worth less than a dollar, we learned to exclude

both the call and put side, not just the side that is less than a dollar. This gives the agent the opportunity to explore both calls and puts on any given day. The more crucial realization was that we could not remove contracts that were initially worth more than a dollar but later dropped to less than a dollar. If we do so, the agent might end up buying a contract that it cannot sell at a later date. We solved this problem by having an ordered list that is sorted in the sequence of strike, expiration, and date, then removing entries that were not already removed because it was below a dollar at an earlier date. Lastly, we only include contracts that expire within the date range to prevent the agent from exploring trades that explores after February 19, 2016 that it cannot close.

3. Methodology

3.1 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. A simple workflow of this implementation is shown at appendix figure 2.

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

3.1.1 State

State consists of the date and a list of the position currently holding. At the start of every episode, the agent will start with state consisting date January 4, 2016 and empty list of position. The total number of possible states for the dataset is enormous. Therefore after each transition, a new state may be added to the state space if the date and position list pair (represented as a tuple) does not already exist. States will be retained for the next episode and will not be removed.

3.1.2 Action

Each state is allowed to buy any available strikes on the current date given the restrictions below. After each transition, a ”sell” action will be added to the action space of the next state. We utilized ε - greedy exploration for choosing the next action. The agent’s possible actions are restricted to a number of restriction. First, the agent is limited to buying one contract per day but able to sell any amount of contracts. Allowing more than single quantity purchases has an astronomical increase on the possible state and action space and complexity. In addition, the agent may not sell contracts that were bought on the same day simply because intraday pricing data is not available. Furthermore, the agent may not write contracts i.e. it can only sell contracts that it’s currently holding. Lastly, the agent may not exercise contracts for stocks. This reduces the possible actions to buying single call or put and closing them for each date.

3.1.3 Transition

After an action is taking, the agent transitions to the next state based on the incremented date and list of positions. If the calendar runs out i.e. the agent reached February 19, 2019, then a new episode begins with starting state on January 4, 2016 holding no positions. After each transition, we will check if any currently held contracts expired. Furthermore, a list of banned actions is maintained in case that the agent chose an action is that not possible e.g. selling an expired contract or not currently held.

3.1.4 Reward

The agent's reward after each action is simply the calculated profit in dollars from selling currently held options. Note that this is not the percentage profit. This plays an important role in shaping the learned optimal strategy. If an option expires, the premium paid for the expire option is the negative reward.

3.1.5 Parameters

We have tested a range of α values with numerous trials. Nearly all runs with low α values have very low convergence rate. Low α trials will cause the agent to be heavily fixated on initial selected actions. Together with high initial ε, trials will vary dramatically based on what the agent randomly explored at the beginning. Similarly, an α value closer 1 will also cause the agent to learn different strategy for each run. We found that a learning rate 0.5 gave the most consistent results.

We also tested a range of values for discount factor. Having a lower discount factor makes the agent more optimal, but takes more episodes to learn. A lower discount factor also implies that the agent will be greedier and prioritize immediate rewards. As seen in appendix figure 3, a discount factor of 0.2 resulted in the agent not improving total Q-value over 30,000 episodes while discount factor of 0.9 allows the agent to learn higher Q-values much more rapidly. We decided to set the medium discount factor of 0.5 to allow for a quicker convergence rate.

ε - greedy exploration enables the agent to occasionally choose a random action instead of one that maximizes the Q-value. This prevents overfitting and improves optimality. As an episode increases, emphasis should be placed on exploitation. The ε - greedy function $1/\sqrt{episode_i}$ allows for a gradual ε reduction.

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 in appendix figure 5.

4. Evaluation

After around 30,000 episodes with the aforementioned Q-value update parameters and restrictions, the agent has derived a greedy trading strategy that focuses on securing immediate profit. By evaluating appendix figure 4 and the transaction history on appendix table 2, we can conclude that these are optimal strategies:

- **Buy deep-in-the-money options during volatile periods** – To adapt to the restriction of buying single quantity option contract per day and that the reward function is based on dollar amount profits instead of a percentage, the agent learned to trade deep-in-the-money options that carry significantly more premium and a higher dollar return proposition. In addition, volatile periods increases the prices of contracts for every strike proportionally, further increasing the profitability of deep-in-the-money options.

- **Buy lowest premium options during flat trends** – On the transaction table, there are some trades that generates small negative returns that the agent found optimal. This unexpected behavior was actually a clever strategy that the agent learned in order to comply with the restrictions. During periods of minimal pricing movement, prices for both calls and puts will all drop due to time value decay. The optimal action would be to hold, but the agent's available action every day is to either buy or sell options. Unable to hold all current position for the day, the 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 and thus higher profitability.

- **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.

5. Conclusion

With reinforcement learning methods such as Q-learning, we were 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. However, it is difficult to verify whether the transactions were truly optimal. To create a baseline would mean brute forcing all the possible trades to generate the true optimal trading strategy so that we can compare it with the agent's performance. Nonetheless, after reviewing the transaction history, we can conclude that the agent's learned strategy is certainly close to being optimal because it follows closely to real world high risk high reward option pricing strategies.

One of our major setbacks was limited computation power 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. Additionally, selecting the learning rate, discount factor, and ε function was a time-consuming task. The effectiveness of these parameters seemed to heavily rely on 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. Lastly, 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.

The agent can be further developed by presenting a better parameter selection from more informative option pricing metrics such as the Greeks and implied volatility. Deep Q-learning is another method that can be used to conjugate the large dataset and reduce computation.

6. Milestones

Date	**Task**	**Progress**
1/23/2019	Submit a project proposal. (1page)	Done
02/06/2019	Collect dataset. Begin baseline models.	Done
02/15/2019	Complete baseline and submit a progress report. Start on redefined baseline model.	Done
02/22/2019	Complete redefined baseline model.	Done
02/27/2019	Complete evaluation metrics	Done
03/06/2019	Complete poster and start on the final report	Done
03/14/2019	Poster presentation	Done
03/17/2019	Submit final project report	Done

References

1. Molly Zhang. SPY option historical pricing dataset from 2005 to 2017. SPY option 2005-2017.csv
2. 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/
3. 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
4. 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
5. Skymind. *A Beginner's Guide to Deep Reinforcement Learning*. Retrieved January 23, 2019, from https://skymind.ai/wiki/deep-reinforcement-learning
6. Martin, K.S. University of the Witwatersrand, Johannesburg.(2014). *Reinforcement Learning Applied to Option Pricing*. Retrieved from https://core.ac.uk/download/pdf/39674013.pdf
7. Folger, Jean. Investopedia. (2017). *Interpreting a Strategy Performance Report*. Retrieved from https://www.investopedia.com/articles/fundamental-analysis/10/strategy-performance-reports.asp
8. Nath, Trevir. Investopedia. (2018). *How Investment Risk Is Quantified.* Retrieved from https://www.investopedia.com/articles/investing/032415/how-investment-risk-quantified.asp
9. ADL. freeCodeCamp. (2018). *An introduction to Q-Learning: reinforcement learning*. Retrieved from https://medium.freecodecamp.org/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc
10. seaotternerd. Stackoverflow. (2015). *Difference bewtween Q-learning and Value Iteration*. Retrieved from https://stackoverflow.com/questions/28937803/difference-between-q-learning-and-value-iteration?newreg=ed6962c3758c49f98e93312c61bd689e
11. Singh, Gaurav. Medium. (2018). *Deep Reinforcement Learning for Algorithmic Trading*. Retrieved from https://medium.com/@gaurav1086/machine-learning-for-algorithmic-trading-f79201c8bac6

12. 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/
13. 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

Appendix: Tables and Images

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.

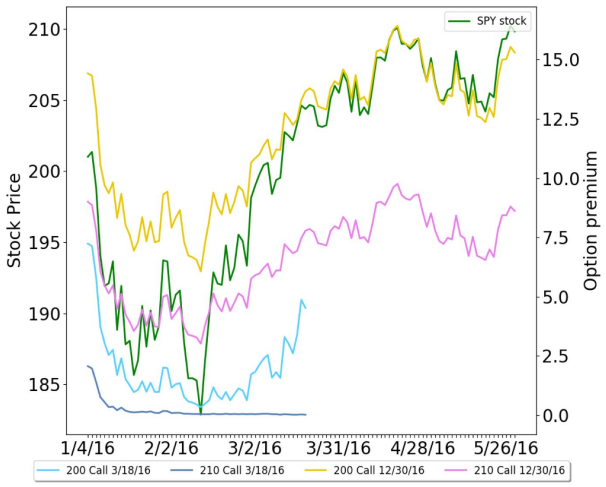

Figure 1. Option Pricing Trend SPY January to June 2016. Moneyness and days until expiration plays a major role in how option pricing fluctuates. This figure is a comparison in how the premium of options with different strike and expiration dates differs over time and stock prices

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 loses its time value. As expiration day approaches, the decay accelerates.

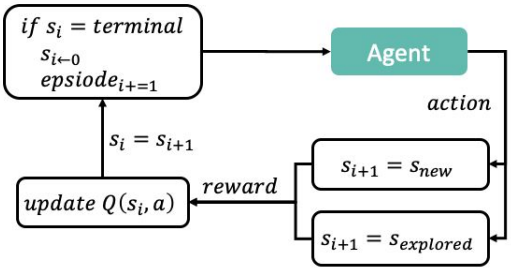

Figure 2. An simple workflow of Q-learning implementation with dynamically added states and actions.

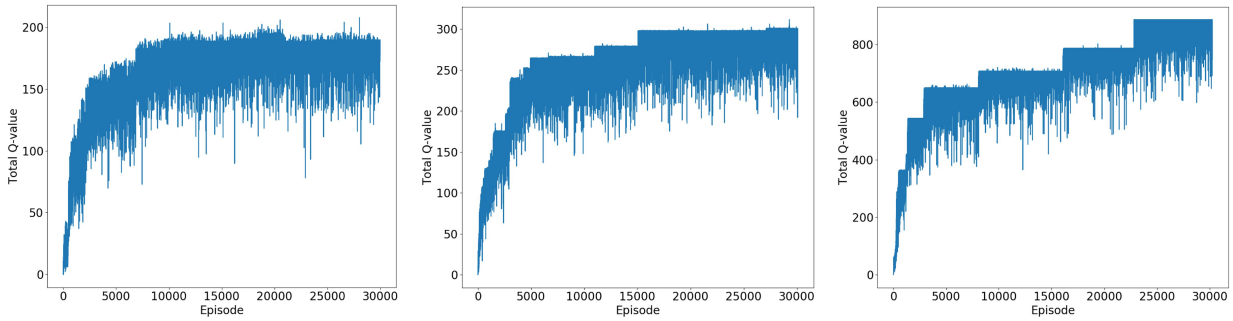

Figure 3. Total Q-value convergence trend across different discount factor γ = 0.2 (left), γ = 0.5 (center), γ = 0.9 (right). A fixed learning rate of 0.5 is used.

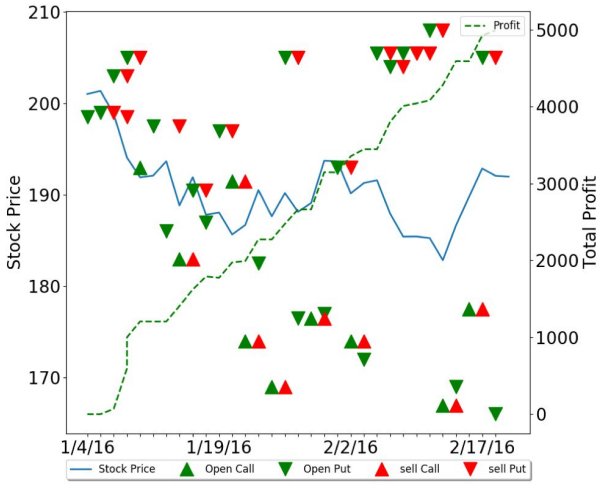

Figure 4. A simulation of the optimal trading strategy was ran and the agent was able to achieve $4991 profit from 1/4/2016 to 2/19/2016. Notice some profit dips during days when SPY is trading flat. Much higher profit is possible if the agent was not limited to single quantity trades. This however brings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

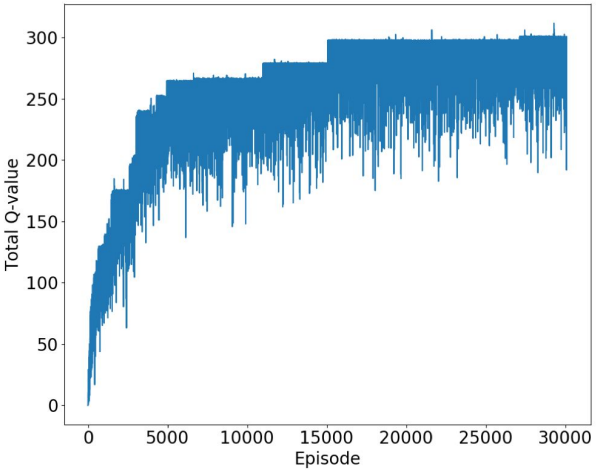

Figure 5. Convergence is consider reached when the difference between current Q-values for all states and the previous Q-values is less than 0.000001. At episode 10,000, ε is 0.01 based on our ε - greedy function, which means that the agent still has 1% chance of exploration. This is shown by the sharp 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				

Table 2. Transaction history of learned optimal trading strategy for January 4, 2016 to February 19, 2016