

# Pricing American Options with Reinforcement Learning

Ashwin Rao

ICME, Stanford University

January 28, 2020

- 1 Review of Optimal Stopping and its MDP Formulation
- 2 Longstaff-Schwartz Algorithm
- 3 RL Algorithms for American Option Pricing: LSPI and FQI
- 4 Choice of feature functions and Model-based adaptation

# Stopping Time

- Stopping time  $\tau$  is a “random time” (random variable) interpreted as time at which a given stochastic process exhibits certain behavior
- Stopping time often defined by a “stopping policy” to decide whether to continue/stop a process based on present position and past events
- Random variable  $\tau$  such that  $Pr[\tau \leq t]$  is in  $\sigma$ -algebra  $\mathcal{F}_t$ , for all  $t$
- Deciding whether  $\tau \leq t$  only depends on information up to time  $t$
- Hitting time of a Borel set  $A$  for a process  $X_t$  is the first time  $X_t$  takes a value within the set  $A$
- Hitting time is an example of stopping time. Formally,

$$T_{X,A} = \min\{t \in \mathbb{R} | X_t \in A\}$$

eg: Hitting time of a process to exceed a certain fixed level

# Optimal Stopping Problem

- Optimal Stopping problem for Stochastic Process  $X_t$ :

$$V(x) = \max_{\tau} \mathbb{E}[G(X_{\tau}) | X_0 = x]$$

where  $\tau$  is a set of stopping times of  $X_t$ ,  $V(\cdot)$  is called the Value function, and  $G$  is the Reward function.

- Note that sometimes we can have several stopping times that maximize  $\mathbb{E}[G(X_{\tau})]$  and we say that the optimal stopping time is the smallest stopping time achieving the maximum value.
- Example of Optimal Stopping: Optimal Exercise of American Options
  - $X_t$  is stochastic process for underlying security's price
  - $x$  is underlying security's current price
  - $\tau$  is set of exercise times corresponding to various stopping policies
  - $V(\cdot)$  is American option price as function of underlying's current price
  - $G(\cdot)$  is the option payoff function, adjusted for time-discounting

# Optimal Stopping Problems as Markov Decision Processes

- We formulate Stopping Time problems as Markov Decision Processes
- *State* is  $X_t$
- *Action* is Boolean: Stop or Continue
- *Reward* always 0, except upon Stopping (when it is  $= G(X_\tau)$ )
- *State*-transitions governed by the Stochastic Process  $X_t$
- For discrete time steps, the Bellman Optimality Equation is:

$$V^*(X_t) = \max(G(X_t), \mathbb{E}[V^*(X_{t+1})|X_t])$$

- For finite number of time steps, we can do a simple backward induction algorithm from final time step back to time step 0

# Mainstream approaches to American Option Pricing

- American Option Pricing is Optimal Stopping, and hence an MDP
- So can be tackled with Dynamic Programming or RL algorithms
- But let us first review the mainstream approaches
- For some American options, just price the European, eg: vanilla call
- When payoff is not path-dependent and state dimension is not large, we can do backward induction on a binomial/trinomial tree/grid
- Otherwise, the standard approach is [Longstaff-Schwartz algorithm](#)
- Longstaff-Schwartz algorithm combines 3 ideas:
  - Valuation based on Monte-Carlo simulation
  - Function approximation of continuation value for in-the-money states
  - Backward-recursive determination of early exercise states

# Ingredients of Longstaff-Schwartz Algorithm

- $m$  Monte-Carlo paths indexed  $i = 0, 1, \dots, m - 1$
- $n + 1$  time steps indexed  $j = n, n - 1, \dots, 1, 0$  (we move back in time)
- Infinitesimal Risk-free rate at time  $t_j$  denoted  $r_{t_j}$
- Simulation paths of prices of underlying as input 2-dim array  $SP[i, j]$
- At each time step,  $CF[i]$  is PV of current+future cashflow for path  $i$
- $s_{i,j}$  denotes state for  $(i, j) := (\text{time } t_j, \text{price history } SP[i, : (j + 1)])$
- $\text{Payoff}(s_{i,j})$  denotes Option Payoff at  $(i, j)$
- $\phi_0(s_{i,j}), \dots, \phi_{r-1}(s_{i,j})$  represent feature functions (of state  $s_{i,j}$ )
- $w_0, \dots, w_{r-1}$  are the regression weights
- Regression function  $f(s_{i,j}) = w \cdot \phi(s_{i,j}) = \sum_{l=0}^{r-1} w_l \cdot \phi_l(s_{i,j})$
- $f(\cdot)$  is estimate of continuation value for in-the-money states

# The Longstaff-Schwartz Algorithm

**Algorithm 2.1:** LONGSTAFFSCHWARTZ( $SP[0 : m, 0 : n + 1]$ )

**comment:**  $s_{i,j}$  is shorthand for state at  $(i,j) := (t_j, SP[i, : (j + 1)])$

$CF[0 : m] \leftarrow [Payoff(s_{i,n}) \text{ for } i \text{ in range}(m)]$

**for**  $j \leftarrow n - 1$  **downto** 1

**do**  $\left\{ \begin{array}{l} CF[0 : m] \leftarrow CF[0 : m] * e^{-r_{t_j}(t_{j+1} - t_j)} \\ X \leftarrow [\phi(s_{i,j}) \text{ for } i \text{ in range}(m) \text{ if } Payoff(s_{i,j}) > 0] \\ Y \leftarrow [CF[i] \text{ for } i \text{ in range}(m) \text{ if } Payoff(s_{i,j}) > 0] \\ w \leftarrow (X^T \cdot X)^{-1} \cdot X^T \cdot Y \\ \textbf{comment:} \text{ Above regression gives estimate of continuation value} \\ \textbf{for } i \leftarrow 0 \textbf{ to } m - 1 \\ \quad \textbf{do } CF[i] \leftarrow Payoff(s_{i,j}) \textbf{ if } Payoff(s_{i,j}) > w \cdot \phi(s_{i,j}) \end{array} \right.$

$exercise \leftarrow Payoff(s_{0,0})$

$continue \leftarrow e^{-r_0(t_1 - t_0)} \cdot mean(CF[0 : m])$

**return** ( $\max(exercise, continue)$ )



# RL as an alternative to Longstaff-Schwartz

- RL is straightforward if we clearly define the MDP
- *State* is [Current Time, History of Underlying Security Prices]
- *Action* is Boolean: Exercise (i.e., Stop) or Continue
- *Reward* always 0, except upon Exercise (= Payoff)
- *State*-transitions governed by Underlying Price's Stochastic Process
- Key is function approximation of state-conditioned continuation value
- Continuation Value  $\Rightarrow$  Optimal Stopping  $\Rightarrow$  Option Price
- We outline two RL Algorithms:
  - Least Squares Policy Iteration (LSPI)
  - Fitted Q-Iteration (FQI)
- Both Algorithms are batch methods solving a linear system
- More details in [Li, Szepesvari, Schuurmans paper](#)

# Least Squares Policy Iteration for Continuation Value

**Algorithm 3.1:** LSPI-AMERICANPRICING( $SP[0:m, 0:n+1]$ )

**comment:**  $s_{i,j}$  is shorthand for state at  $(i,j) := (t_j, SP[i:(j+1)])$

**comment:**  $A$  is an  $r \times r$  matrix,  $b$  and  $w$  are  $r$ -length vectors

**comment:**  $A_{i,j} \leftarrow \phi(s_{i,j}) \cdot (\phi(s_{i,j}) - \gamma \mathbb{I}_{w \cdot \phi(s_{i,j+1}) \geq \text{Payoff}(s_{i,j+1})} * \phi(s_{i,j+1}))^T$

**comment:**  $b_{i,j} \leftarrow \gamma \mathbb{I}_{w \cdot \phi(s_{i,j+1}) < \text{Payoff}(s_{i,j+1})} * \text{Payoff}(s_{i,j+1}) * \phi(s_{i,j})$

$A \leftarrow 0, B \leftarrow 0, w \leftarrow 0$

**for**  $i \leftarrow 0$  **to**  $m-1$

**do**  $\left\{ \begin{array}{l} \text{for } j \leftarrow 0 \text{ to } n-1 \\ \quad \text{do } \left\{ \begin{array}{l} Q \leftarrow \text{Payoff}(s_{i,j+1}) \\ P \leftarrow \phi(s_{i,j+1}) \text{ if } j < n-1 \text{ and } Q \leq w \cdot \phi(s_{i,j+1}) \text{ else } 0 \\ R \leftarrow Q \text{ if } Q > w \cdot P \text{ else } 0 \\ A \leftarrow A + \phi(s_{i,j}) \cdot (\phi(s_{i,j}) - e^{-r t_j(t_{j+1}-t_j)} * P)^T \\ B \leftarrow B + e^{-r t_j(t_{j+1}-t_j)} * R * \phi(s_{i,j}) \end{array} \right. \\ w \leftarrow A^{-1} \cdot b, A \leftarrow 0, b \leftarrow 0 \text{ if } (i+1) \% \text{BatchSize} == 0 \end{array} \right.$

# Fitted Q-Iteration for Continuation Value

**Algorithm 3.2:** FQI-AMERICANPRICING( $SP[0 : m, 0 : n + 1]$ )

**comment:**  $s_{i,j}$  is shorthand for state at  $(i, j) := (t_j, SP[i, : (j + 1)])$

**comment:**  $A$  is an  $r \times r$  matrix,  $b$  and  $w$  are  $r$ -length vectors

**comment:**  $A_{i,j} \leftarrow \phi(s_{i,j}) \cdot \phi(s_{i,j})^T$

**comment:**  $b_{i,j} \leftarrow \gamma \max(\text{Payoff}(s_{i,j+1}), w \cdot \phi(s_{i,j+1})) * \phi(s_{i,j})$

$A \leftarrow 0, B \leftarrow 0, w \leftarrow 0$

**for**  $i \leftarrow 0$  **to**  $m - 1$

**do**  $\left\{ \begin{array}{l} \text{for } j \leftarrow 0 \text{ to } n - 1 \\ \quad \text{do } \left\{ \begin{array}{l} Q \leftarrow \text{Payoff}(s_{i,j+1}) \\ P \leftarrow \phi(s_{i,j+1}) \text{ if } j < n - 1 \text{ else } 0 \\ A \leftarrow A + \phi(s_{i,j}) \cdot \phi(s_{i,j})^T \\ B \leftarrow B + e^{-r t_j (t_{j+1} - t_j)} * \max(\text{Payoff}(s_{i,j+1}), w \cdot P) * \phi(s_{i,j}) \end{array} \right. \\ w \leftarrow A^{-1} \cdot b, A \leftarrow 0, b \leftarrow 0 \text{ if } (i + 1) \% \text{BatchSize} == 0 \end{array} \right.$

# Feature functions

- Li, Szepesvari, Schuurmans recommend Laguerre polynomials (first 3)
- Over  $S' = S_t/K$  where  $S_t$  is underlying price and  $K$  is strike
- $\phi_0(S_t) = 1, \phi_1(S_t) = e^{-\frac{S'}{2}}, \phi_2(S_t) = e^{-\frac{S'}{2}} \cdot (1 - S'), \phi_3(S_t) = e^{-\frac{S'}{2}} \cdot (1 - 2S' + S'^2/2)$
- They used these for Longstaff-Schwartz as well as for LSPI and FQI
- For LSPI and FQI, we also need feature functions for time
- They recommend
$$\phi_0^t(t) = \sin\left(\frac{\pi(T-t)}{2T}\right), \phi_1^t(t) = \log(T-t), \phi_2^t(t) = \left(\frac{t}{T}\right)^2$$
- They claim LSPI and FQI perform better than Longstaff-Schwartz with this choice of features functions
- We will code up these algorithms to validate this claim ☺