Deep Optimal Stopping & Pricing of American-Style Options

Applied Quantitative Finance Seminar

Sabina Georgescu, Konrad Müller

Overview

---

1. Introduction to Bermudan Options

Context

- European Option: Exercisbale on a **single** date
- Bermudan Option: Exercisable on **several** dates
- American Option: Exercisable on **any** date before expiration

$\therefore \quad$ Bermudan contracts lie between European & American

---

Bermudan Max-Call Option

- Call Option on the maximum of d underlying assets $S^{1,...,d}$
- Finitely many exercise dates $0 = t_0 < t_1 < ...< t_N = T$
- Discounted payoff at exercise time t_n:

$$ G_{t_n} =e^{-r t_n}\max_{i = 1, ..., d} \left(S_{t_n}^i - K\right)^+ $$

Pricing & Dynamics

- Multi-dimensional Black-Scholes model under risk-neutral dynamics

- d-dimensional Brownian Motion W with uncorrelated instantaneous components:

$$ dS^i_t = S_0^i \exp{\left(\left[r - \delta - \frac{\sigma^2}{2}\right]dt + \sigma dW^i_t\right)}$$

- Pricing formula with $\sup$ attained at τ_n:

$$ V_{t_n} = \sup_{\tau\in{t_n,...,t_N}} \mathbb{E}[G_\tau \mid \mathcal{F}_{t_n}] $$

---

2. Least-Squares Monte Carlo

Longstaff & Schwartz, *Valuing American Options by Simulation: A Simple Least-Squares Approach* (2001)

LSM Method

Estimate **continuation values** by regressing simulated discounted cash flows on the linear span of finitely many basis functions $L_0,...,L_B$ of ITM paths:

$$ F(\omega; t_n) \approx \sum_{i=0}^B w^_i \cdot L_i(X(\omega; t_n)) \quad\text{for}\quad \textbf{w}^ = \min_{w\in\mathbb{R}^{B+1}} || \textbf{Y}{t_n} - B(t_n)\cdot\textbf{w}\|{L_2} $$

where the discounted cash flows vector and basis functions matrix are given by: $$\textbf{Y}_{t_n} = \left[Y(\omega_0; t_n),...,Y(\omega_m; t_n)\right]^T,$$ $$ \textbf{B}_m(t_n) = \left[L_0(X(\omega_m;t_n)),...,L_B(X(\omega_m;t_n))\right]^T, \forall m\in{1,...,M}.$$

Optimal Stopping Value captured with the backwards recursion:

$$ \textit{Snell Envelope}\quad V_{t_n} = \begin{dcases} G_T \quad\text{if } n=N,\ \max {G_{t_n}, F_{t_n}} \quad\text{if } n<N; \end{dcases} $$

Optimal Stopping Rule: $$ \textit{Dynamic Programming Eqn.} \quad\tau_n = \begin{dcases} t_n \quad\text{if } G_{t_n} \ge F_{t_n},\ \tau_{n+1} \quad\text{if } G_{t_n} < F_{t_n}; \end{dcases} $$

$\therefore$ Bermudan Max-Call valuation: $$ V_{LSM} := \sum_{i=1}^{m}V_{t_0}(\omega_m).$$

Continuation Values

- Conditional expected payoffs obtained from continuing the option
- Borel-measurable functions in the **Hilbert space** $L^2(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{F})$
- $L^2(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{F})$ admits **orthonormal bases**, e.g. Laguerre polynomials
- Write the continuation values in such a basis

$\therefore$ **OLS-regression** with **polynomial features** on the linear span of the chosen basis

---

Choice of Basis Functions

Regression Features for multi-asset Bermudan options: not restricted to basis functions

- Longstaff-Schwartz (2001), $d=5$ asets:
  - First 5 Hermite polynomials in the value of the most expensive asset
  - Value and square of the other 4 assets
  - Selected products between individual asset prices
- Broadie & Cao (2008):
  - Between $12 - 18$ polynonmial basis functions, up to 5^{th} order

---

LSM Algorithm: Pseudo-Code

```
def lsm(paths):

    payoff_at_stop = payoff_fn(N, paths[:, -1])

    for n in np.arange(start=N, stop=0, step=-1):

        x_n = paths[:, n]
        payoff_now = payoff_fn(n, x_n)

        features = feature_map(x_n, payoff_now)
        model = LinearRegression()
        model.fit(features, payoff_at_stop)
        continuation_values = model.predict(features)

        idx = payoff_now >= continuation_values
        payoff_at_stop[idx] = payoff_now[idx]

    biased_price = payoff_at_stop.mean()
```

Feature Engineering: Issues

- Feature maps not rich enough to accurately price complex options
- Potentially over-engineered to payoff specifics or market dynamics
- **Curse of Dimensionality**: $\#$ basis functions grows "only" polynomially in d, but rapidly becomes infeasible

$\therefore$ Solution: *Learn* the feature maps $\rightarrow$ Deep LSM

---

3. Deep LSM

Becker et al., *Pricing and Hedging American-Style Options with Deep Learning* (2020)

Candidate Optimal Stopping Strategy

Estimate **continuation values** with a feedforward Deep Neural Network. Project $G_{\tau_{n+1}}$ on a subset ${c^\theta(X_{t_n})}_\theta$ of Borel-measurable functions $c^\theta:\mathbb{R}^d\rightarrow\mathbb{R}$ parameterised by θ, as:

$$\mathbb{E}\left[G_{\tau_{n+1}}\mid X_{t_n}\right] = c^\theta(X_{t_n})$$

Learn optimal hyperparameter θ_n by employing SGD to mimise over θ: $$\mathbb{E}\left[\left(G_{\tau_{n+1}} - c^\theta(X_{t_n})\right)^2\right]$$

$\therefore$ Summarise all continuation values along the path with: $$\Theta := \left(\theta_0,...,\theta_N\right).$$

Optimal Stopping Value via the *Snell Envelope*: $$ V_{t_n} = \begin{dcases} G_T \quad\text{if } n=N,\\ \max {G_{t_n}, c^{\theta_n}(X_{t_n})} \quad\text{if } n<N; \end{dcases} $$

Optimal Stopping Rule via the *Dynamic Programming Equation*: $$ \tau_n = \begin{dcases} t_n \quad\text{if } G_{t_n} \ge c^{\theta_n}(X_{t_n}),\\ \tau_{n+1} \quad\text{if } G_{t_n} <c^{\theta_n}(X_{t_n}); \end{dcases} $$

$\therefore$ **Optimal Stopping Time**: $$\tau^\Theta := \min {n\in{0,...,N} \mid G_{t_n} \ge c^{\theta_n}(X_{t_n})}.$$

Lower Bound

Lower Bound estimate: $$L \approx \mathbb{E} \left[G_{\tau^\Theta}\right]$$

where $G_{\tau^\Theta}\approx g^k$ by the Optimal Stopping Rule τ^Θ applied to a further K_L independently generated underlying paths. Approximate L with Monte Carlo averaging.

$\therefore$ **Lower Bound**: $$ \hat{L} = \frac{1}{K_U} \sum_{k=K+1}^{K+K_L} g^k.$$

Upper Bound

Upper Bound estimate via **Doob-Mayer Decomposition Theorem**:

$$ U \approx \mathbb{E} \left[\max_{0\le n \le N} \left(G_{t_n} - M^\Theta_{t_n} -\epsilon_n\right)\right]. $$

Refer to *Deep Optimal Stopping* for the nested simulation, with another K_U independently generated underlying paths, of the martingale realisations $m^k_n.$ Monte Carlo average:

$\therefore$ **Upper Bound**: $$ \hat{U} = \frac{1}{K_U} \sum_{k=K+K_L+1}^{K+K_L+K_U} \max_{i\le n\le N} (g^k_n - m^k_n). $$

Point Estimate & Confidence Interval

Point Estimate: $$ \hat{V} = \frac{\hat{L}+\hat{U}}{2}$$

Sample **standard deviations** of the bounds by the **Central Limit Theorem**:

$$\hat\sigma_L = \sqrt{\frac{1}{K_L-1} \sum_{k=K+1}^{K+K_L} \left(g^k-\hat{L}\right)^2},$$

$$\hat\sigma_U = \sqrt{\frac{1}{K_U-1} \sum_{k=K+K_L+1}^{K+K_L+K_U} \left(\displaystyle\max_{0\le n\le N}\left(g(n,x^k_n) - m^k_n\right)-\hat{U}\right)^2}.$$

Confidence Interval: By the CLT, the Optimal Stopping Value from the Snell admits the asymptotically valid two-sided $1-\alpha$ interval:

$$\left[\hat{L} - z_{\alpha/2} \frac{\hat\sigma_L}{\sqrt{K_L}}, \hat{U} + z_{\alpha/2} \frac{\hat\sigma_U}{\sqrt{K_U}}\right]$$

where $z_{\alpha/2}$ is the $(1-\alpha/2)^{th}$ quantile of the standard Gaussian.

---

4. Deep Optimal Stopping

Becker et al., *Deep Optimal Stopping* (2019a)

Motivation

- **Deep Learning** solution to the **Optimal Stopping Problem**, circumventing the traditional estimation of continuation values $\rightarrow$ DOS
- The immediate payoff & continuation value **comparison** is **inherent** through the choice of **reward/loss** function, thus performed indirectly
- Method based on an **explicit parameterisation** of the **stopping decision**
- Decompose the **optimal stopping rule** in a sequence of **binary decisions**, estimated recursively with a sequence of feedforward DNNs

---

Parameterising the Stopping Decision

% to be continued

$$ \tau_{n+1} = \sum_{m=n+1}^{N} \left[m \cdot f^{\theta_m}(X_m) \cdot \prod_{j=n+1}^{m-1}(1-f^{\theta_j}(X_j))\right]$$

Parameter Optimisation

- Reward (Loss) calculated using the **soft** stopping decision

$$ r^k_n(\theta) = g(n,x^k_n)\cdot F^\theta(x^k_n) + g(l^k_{n+1},x^k_{l^k_{n+1}})\cdot (1-F^\theta(x^k_n))$$

---

5. Dual LSM

Rogers, *Monte Carlo Valuing of American Options* (2002)

Dual Valuation

% will keep it real short

---

6. Results

Results: LSM

Features	d	S_0	P_{biased}	s.e.	L	U	Point Est.	95 % CI
base	5	90	16.33	0.03	16.54	16.89	16.72	[16.30, 16.93]
ls	5	90	16.56	0.04	16.41	16.57	16.49	[16.17, 16.59]
r-NN	5	90	16.63	0.04	16.56	16.78	16.67	[16.33, 16.81]
base	5	100	25.71	0.06	25.72	26.28	26.00	[25.43, 26.32]
ls	5	100	26.01	0.02	26.03	26.21	26.12	[25.75, 26.24]
r-NN	5	100	25.94	0.04	26.00	26.32	26.16	[25.72, 26.35]

Results: DLSM

d	S_0	L	U	Point Est.	95 % CI
5	90	16.62	16.66	16.64	[16.60, 16.66]
5	100	26.10	26.18	26.14	[26.09, 26.20]
5	110	36.69	36.79	36.74	[36.67, 36.81]
10	90	26.05	26.33	26.19	[26.03, 26.37]
10	100	38.09	38.40	38.24	[38.06, 38.43]
10	110	50.60	50.99	50.80	[50.57, 51.03]

Results: DOS

d	S_0	L	U	Point Est.	95 % CI
5	90	16.60	16.65	16.62	[16.59, 16.66]
5	100	26.11	26.18	26.15	[26.10, 26.19]
5	110	37.78	37.86	37.82	[37.76, 37.88]
10	90	26.14	26.30	26.22	[26.13, 26.33]
10	100	38.23	38.45	38.34	[38.21, 38.49]
10	110	50.95	51.17	51.06	[50.93, 51.22]

Comparison to the Literature

d	S_0	DOS	DLSM	Literature
5	90	16.62	16.64	16.64
5	100	26.15	26.14	26.15
5	110	37.82	36.74	36.78
10	90	26.22	26.19	26.28
10	100	38.34	38.24	38.37
10	110	51.06	50.80	50.90

References

Literature

- Valuing American Options by Simulation: A Simple Least-Squares Approach
- Deep optimal stopping
- Pricing and Hedging American-Style Options with Deep Learning
- Optimal Stopping via Randomized Neural Networks
- Monte Carlo Valuing of American Options

Code

https://github.com/konmue/american_options

Q&A